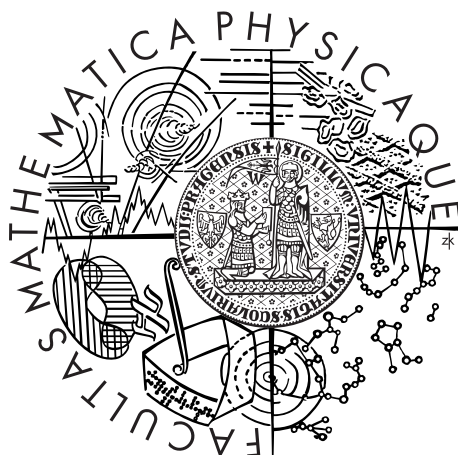


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Pavel Hryzlík

Využití Linked Data pro sdílení dat o smlouvách veřejných institucí

Katedra softwarového inženýrství

Vedoucí diplomové práce: Doc. Mgr. Martin Nečaský, Ph.D.

Studijní program: Informatika

Studijní obor: I2 Softwarové systémy

Praha 2015

Zde bych rád poděkoval vedoucímu práce Doc. Mgr. Martinu Nečaskému, Ph.D. za správné směřování, rady a nápady. Dále bych chtěl poděkovat Ondřeji Profantovi a také PhDr. Ing. Jiřímu Skuhrovcovi. Díky nim jsem mohl lépe proniknout do světa otevřených dat. Díky patří pochopitelně také mým nejbližším.

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Využití Linked Data pro sdílení dat o smlouvách veřejných institucí

Autor: Bc. Pavel Hryzlík

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: Mgr. Martin Nečaský, Ph.D., Katedra softwarového inženýrství

Abstrakt: Cílem diplomové práce je prozkoumat možnosti využití principů Linked Data pro publikaci a sdílení dat o smlouvách veřejných institucí a jejich propojení na související data ve veřejném prostoru (např. obchodní a živnostenský rejstřík, registr veřejných zakázek, apod.). Práce představí kompletní proces otevírání smluv. Definuje datový standard pro otevřené smlouvy a navrhne ontologii pro publikaci dat o smlouvách a jejich propojení. Dále navrhne a implementuje platformu pro publikaci smluv. První částí platformy je konverzní modul umožňující konverzi smluv uložených v relačních databázích do RDF podoby. Využije zde techniky R2RML mapování. Druhou částí je jednotné úložiště stahující údaje o smlouvách v Linked Data podobě. Třetí částí je webová aplikace, která data o smlouvách zpřístupní koncovým uživatelům.

Klíčová slova: Smlouva, Otevřená data, Linked Data, RDF, JSON-LD, R2RML, SPARQL

Title: Exploitation of Linked Data for sharing public agreements data

Author: Bc. Pavel Hryzlík

Department: Department of Software Engineering

Supervisor: Doc. Mgr. Martin Nečaský, Ph.D., Department of Software Engineering

Abstract: The objective of the thesis is to explore the possibilities of using Linked Data principles for publishing and sharing data on contracts of public institutions and their connections to related data in the public domain (eg. Business and trade register, register of contracts, etc.). Thesis presents the entire process of opening up contracts. Defines a data standard for open contracts and proposes an ontology for the publication of data on contracts and their interconnections. Furthermore, it designs and implements a platform for publishing contracts. The first part of the platform is a conversion module enabling the conversion of contracts stored in relational databases into RDF form. Employed are R2RML mapping techniques. The second part is a uniform repository that downloads data on contracts in Linked Data format. The third part is a web application that will make the data on contracts available to end users.

Keywords: Contract, Open Data, Linked Data, RDF, JSON-LD, R2RML, SPARQL

Obsah

1	Úvod	3
1.1	Motivace	4
1.2	Cíl práce	4
1.3	Struktura práce	5
2	Otevřená data a principy Linked Data	6
2.1	Otevřená data (Open Data)	6
2.2	Kvalita otevřených dat	8
2.3	Stupně otevřenosti	8
2.4	Propojitelná data (Linked Data)	10
2.5	Otevřená a propojitelná data (Linked Open Data - LOD)	10
2.6	Výhody a přínosy otevřených dat a principů Linked data	11
2.7	RDF (Resource Description Framework)	12
2.8	RDF Ontologie	14
2.8.1	Propojování se souvisejícími entitami	15
2.9	Publikace	16
2.9.1	Příklad dat serializovaných ve formátu N-Triples	16
2.9.2	Příklad dat serializovaných ve formátu Turtle	17
2.9.3	Příklad dat serializovaných ve formátu JSON-LD	18
3	Otevřené smlouvy	21
3.1	Situace ve veřejné správě ČR	21
3.2	Standard pro zveřejňování smluv	22
3.2.1	Základní struktura	22
3.2.2	Reprezentované entity	25
3.2.3	Číselníky	31
3.3	Publikace	32
3.3.1	JSON	32
3.3.2	CSV	36
3.4	Metodika zveřejňování smluv	39
4	Otevřené smlouvy jako Linked Data	40
4.1	Přiřazení identifikátorů jednotlivým entitám otevřených smluv	40
4.2	Ontologie pro publikaci dat o smlouvách	42
4.2.1	Analýza vhodných, již existujících ontologií	42
4.2.2	Tvorba ontologie	45
4.2.3	Publikace	50
4.3	Možnosti propojení na související data	51
4.4	Provázání s datovým formátem JSON	51
5	Požadavky na platformu pro otevřené smlouvy	58
5.1	Funkční požadavky	58
5.2	Nefunkční požadavky	59

6	Návrh platformy pro otevřené smlouvy	60
6.1	Architektura	60
6.2	Konverzní mechanismus	62
6.3	Jednotné úložiště	63
6.4	Datová síť	65
6.5	Webová aplikace	66
7	Implementace platformy	68
7.1	Konverzní mechanismus	68
7.1.1	Munis ESML	68
7.1.2	R2RML mapování	70
7.1.3	Volba technologií a implementační platformy	81
7.1.4	SPARQL endpoint	82
7.1.5	Zpracování RDF výstupu	83
7.1.6	Konfigurace	84
7.1.7	Požadavky na architekturu	84
7.2	Jednotné úložiště	85
7.2.1	Nástroj Unified views	85
7.2.2	Požadavky na architekturu	88
7.3	Webová aplikace	88
7.3.1	Volba technologií a implementační platformy	88
7.3.2	Získávání dat	89
7.3.3	Požadavky na architekturu	97
8	Evaluace	98
8.1	Test konverzního modulu	98
8.2	Test webové aplikace	101
9	Shrnutí procesu otevírání smluv	102
	Závěr	104
	Seznam zdrojů a použité literatury	106
	Seznam obrázků	111
	Seznam tabulek	112
	Výpisy kódu	113
	Přílohy	114
	A Příloha	115
	B Příloha	116
	C Příloha	117
	D Příloha	118
	E Příloha	124

1. Úvod

V době informační společnosti se využívání internetu stalo naší každodenní rutinou. Skrze různé webové aplikace a služby každodenně pracujeme s obrovským množstvím informací. Běžně komunikujeme přes e-mail, finance spravujeme skrze internetové bankovníctví, část svého osobního života sdílíme na sociálních sítích. Požadavek na on-line vyřizování agendy vůči veřejné správě tedy není překvapujícím.

Problematika elektronizace veřejné správy, jednotně nazývaná jako „e-government“, je aktuálním tématem již po mnoho let. Důsledkem tohoto procesu je generování obrovského množství nesmírně důležitých dat. Tato data ale v naprosté většině případů leží schovaná v databázích jednotlivých veřejných institucí. Mnoho z těchto dat by ale ze zákona mělo být volně dostupných. Často však jediným možným způsobem, jak taková data získat je použití zákona č.106/1999 Sb.[1], o svobodném přístupu k informacím. Netřeba zmiňovat, že tato snaha se mnohdy může stát značně netriviální.

Řešením je vhodná data, resp. metadata o těchto datech, zpřístupnit on-line. Pro strojově čitelná data zveřejněná na internetu se zažil pojem Otevřená data. Tato data pak může vyhledávat a zpracovávat kdokoli. To přináší řadu dílčích výhod od úspory nákladů, přes boj s korupcí, až po zapojení občanů, nemluvě o podnikatelském potenciálu, převážně možnosti vzniku mnoha užitečných aplikací pracujících nad otevřenými daty. To celé za cenu minimálních nákladů z veřejných rozpočtů.

Otvírání dat můžeme chápat jako další krok v procesu elektronizace veřejné správy. Průkopníky v této oblasti jsou státy s vyspělou formou demokracie, jako USA a Spojené království. Příklad si ale také můžeme vzít od Estonska. Malá země, vědoma si, že nemá nerostné bohatství ani rozvinutý průmysl, se rozhodla prosadit na poli informačních technologií, kde základem jsou otevřené on-line služby veřejné správy. Důležitost otevřených dat si uvědomuje i Evropská unie. Směrnicí 2013/37/EU[2] v podstatě doporučuje členským státům, aby data otvíraly. České republice se také povedlo nastartovat procesy otvírání veřejné správy. Pokrok je cítit hlavně na národní úrovni. Mezi městy a obcemi jsou však otevřená data často stále neznámým pojmem. Problematikou a obecně osvětou otevřených dat se zabývá mimo jiné Ministerstvo vnitra ČR[3], projekt Rekonstrukce státu[4], Fond Otakara Motejla[5], Oživení o.s.[6], Fórum pro otevřená data[7], či iniciativa OpenData.cz[8].

Otevřená data však nelze chápat jako samospásné řešení problémů veřejné správy. Jsou spíše prostředkem ke zvýšení otevřenosti a transparentnosti. Veřejná služba však může být netransparentní i s otevřenými daty. Řekněme, že pro kvalitní veřejnou službu jsou otevřená data nutnou, nikoli však postačující podmínkou.

Dalším aspektem otevřených dat je jejich kvalita. Kvalitní otevřená data jsou propojena mezi sebou v rámci jednotného sdíleného prostoru, mohou na sebe odkazovat a využívat širokého kontextu, které takový sdílený prostor propojených dat nabízí. Taková data využívají principů Linked Data.[10, 11, 12, 13]

1.1 Motivace

Základní motivace této práce je rozdělena do tří pilířů:

Veřejnoprávní sféra

Na podzim roku 2014 se konal seminář Transparentnost v obcích[14] v Poslanecké sněmovně pořádaný panem Mgr. Janem Farským. V rámci semináře se sešla skupina složená ze zástupců měst a obcí, akademické sféry a neziskového sektoru. Předmětem jednání byla otevřená data. Výsledkem bylo rozhodnutí, že první datovou sadou vhodnou k plošnému otevření, také vzhledem k chystanému zákonu o registru smluv[9], jsou údaje o smlouvách. Prvním krokem je standardizace datového formátu, resp. určení položek vhodných ke zveřejnění. Motivací bylo, že pokud standard začne využívat netriviální počet měst a obcí, tak je reálná šance k prosazení standardu na národní úroveň. Ustanovila se tedy, pod záštitou Oživení o.s. a EconLabu (dříve Centra aplikované ekonomie o.s.)[15], „akční“ skupina, jejímž cílem byla tvorba datového standardu pro otevřené smlouvy. Bylo mi ctí stát se členem této skupiny.

Komerční sféra

Jako externista se podílím na tvorbě software pro veřejnou správu ve společnosti Triada spol. s.r.o. Mým úkolem se ke konci roku 2014 stala tvorba modulu ESML pro interní evidování smluv.

Akademická sféra

V rámci MFF UK ve spolupráci s Fakultou informatiky VŠE vznikla iniciativa OpenData.cz. Jejím cílem je vybudování otevřené datové infrastruktury v České republice. Na MFF UK také probíhá výzkum propojitelných dat, Linked Data. Mým cílem bylo přispět k otevřené datové infrastruktuře, navíc s využitím principů Linked Data. Rozhodnutí věnovat se publikaci dat o smlouvách padlo již v červnu 2014. Konkrétní obrysy však práce získala až s přispěním výše zmíněných pilířů.

Výsledkem je tedy aplikace principů Linked Data pro publikaci a sdílení dat o smlouvách s možností konkrétního využití nad modulem ESML společnosti Triada. To celé s ohledem na vznikající datový standard. Jednou z důležitých motivací bylo, že v případě prosazení datového standardu na národní úroveň mohou města a obce používající modul ESML využitím této práce automaticky zveřejňovat smlouvy v Linked Data podobě, a to s minimálními náklady. Taková data lze pak agregovat do jednotných úložišť, nad kterými mohou vznikat nejrůznější aplikace přinášející konečný přínos pro uživatele.

1.2 Cíl práce

Cílem práce je prozkoumat možnosti využití principů Linked Data pro publikaci a sdílení dat o smlouvách veřejných institucí a jejich propojení na související data ve veřejném prostoru. Prvním krokem je definování datového standardu a

ontologie pro otevřené smlouvy. Dalším krokem je návrh způsobu konverze dat stávajícími informačními systémy veřejných institucí (v podobě relačních databází) do otevřeného formátu využívající principy Linked Data a implementace konverzního mechanismu pro vybraný konkrétní informační systém (Triada spol. s.r.o). V dalším kroku následuje návrh a implementace jednotného úložiště dat o smlouvách v Linked Data s experimentálním zprovozněním na serveru poskytnutém vedoucím práce. V jednotném úložišti se očekává návrh řešení integračních problémů dané heterogenitou dat publikovaných různými veřejnými institucemi. Následujícím krokem je nad tímto jednotným úložištěm návrh a implementace webové aplikace, která data o smlouvách zpřístupní koncovým uživatelům.

1.3 Struktura práce

Obsah práce je rozdělen na 10 kapitol a 5 příloh. Ve druhé kapitole jsou popsány a vysvětleny základní principy otevřených dat. Třetí kapitola se zabývá pojmem otevřené smlouvy. Kapitola nejdříve rozebere aktuální stav otevřenosti smluv ve veřejné správě a následně nastíní vznikající datový standard. Čtvrtá kapitola zadefinuje otevřené smlouvy jako Linked Data. V páté kapitole se definují požadavky na platformu pro otevřené smlouvy. Šestá kapitola zmíněnou platformu navrhne. Sedmá kapitola se zabývá konkrétní implementací platformy. V osmé kapitole jsou znázorněny zátěžové testy některých dílčích částí implementace. Devátá kapitola nastíní proces otevírání smluv formou obecné metodiky. Poslední, desátou kapitolou je závěr shrnující práci jako celek. Nedílnou součástí práce je seznam použité literatury, obrázků, tabulek a výpisů kódů. Práce zahrnuje také 5 příloh. V příloze A je znázorněn harmonogram vývoje standardu otevřených smluv. V příloze B se nachází uživatelská dokumentace. Příloha C popisuje strukturu příloženého datového nosiče. V příloze D se nachází Linked Data ontologie pro otevřené smlouvy. Konečně, v příloze E je R2RML skript mapující tabulky z relační databáze do RDF.

2. Otevřená data a principy Linked Data

Předmětem této kapitoly je čtenáře stručně seznámit se základními pojmy a principy otevřených, propojitelných dat a následně s technologiemi sloužícími k jejich zápisu a zpracování.

2.1 Otevřená data (Open Data)[16, 13]

„Open data can help us address the greatest challenges of our time and generate value for everyone“ - Open Data Institute 2012¹

Začneme definicí, kterou si postupně vysvětlíme. Jako otevřená data můžeme chápat údaje zveřejněná na internetu, která jsou

1. *úplná*
2. *snadno dostupná*
3. *strojově čitelná*
4. *používající standardy s volně dostupnou specifikací*
5. *zpřístupněna za jasně definovaných podmínek užití dat s minimem omezení*
6. *dostupná uživatelům při vynaložení minima možných nákladů*[3]

Úplnost

Pokud se rozhodneme zveřejňovat data, tak v případě, že nás neomezuje zákon, či jiná restriktivní opatření, měli bychom dbát na to, aby byla úplná, resp. v maximálním možné rozsahu. Není cílem zveřejňovat útržky ztrácející vypovídající hodnotu.

Snadná dostupnost

Základní požadavek na dostupnost otevřených dat spočívá v tom, že by měla být k dispozici kdykoli, ne pouze např. na vyžádání. Otevřená data budou také přínosem pro širokou veřejnost jedině tehdy, pokud budou snadno dohledatelná. Skrytá data za změní odkazů se hledají špatně.

Strojová čitelnost

Klíčovou vlastností otevřených dat je strojová čitelnost. Otevřeným datům by měl porozumět nejen člověk, ale i stroj. Účelem je umožnit data automatizovaně zpracovávat, analyzovat, počítat statistiky apod.

¹Logo otevřených dat viz Obr. 2.1

Otevřené standardy

Software, nástroje či metodiky potřebné k zpracování dat by měly být volně dostupné. Data v uzavřeném formátu, která potřebují ke zpracování konkrétní proprietární software, postrádají smysl otevřenosti.

Zpřístupněna za jasně definovaných podmínek

Typicky je třeba dbát na to, aby data byla zveřejňována pod otevřenou licenci.²

Dostupná uživatelům s minimem nákladů

Je třeba si uvědomit, že nezveřejňujeme data pro data. Zveřejňujeme pro přidanou hodnotu, např. pro lepší službu nebo vyšší efektivitu. Náklady na zveřejnění by tak neměly přesáhnout případná zlepšení.



Obrázek 2.1: Logo otevřených dat

²Více k problematice licencování a užití otevřených dat lze dohledat na webu Ministerstva vnitra^[3]

2.2 Kvalita otevřených dat

Tvůrce WWW a ředitel konsorcia W3C Tim Berners-Lee navrhl pěti hvězdičkový systém, jak kategorizovat otevřená data (viz Obr. 2.2). Každá hvězdička definuje stupeň otevřenosti, kde 5★ znamená nejvyšší kvalitu dat, 1★ naopak nejmenší. Také platí, že každý stupeň je nadmnožinou (rozšířením) stupně předešlého.[17, 18]

2.3 Stupně otevřenosti[17, 18]

★ Libovolná zveřejněná data pod otevřenou licencí

- Přínosy pro uživatele - uživatel může data číst, tisknout, ukládat, přenášet, měnit a sdílet podle svého uvážení
- Přínosy/náklady pro vydavatele - velmi nenáročné na publikaci
- Příkladem může být formát PDF

Publikace dat na úrovni 1★ je zdaleka nejjednodušší a nepotřebuje příliš vynaloženého úsilí. Určitě je lepší zveřejňovat data na úrovni 1★, než vůbec. Využitelnost dat však může být velmi obtížná, např. díky nutnosti dolování dat z PDF dokumentů.

★★ Strukturovaná data ve strojově čitelném formátu

- Přínosy pro uživatele - uživatel může pokročile zpracovávat data s využitím proprietárních nástrojů k tomu určených
- Přínosy/náklady pro vydavatele - velmi nenáročné na publikaci
- Příkladem může být formát MS Excel (.xls)

V dnešní době už poměrně rozšířený způsob publikace dat. Zpracování dat ale vyžaduje specifické nástroje k tomu určené. Pokud tedy chceme zpracovávat např. excelovskou tabulku (.xls), potřebujeme k tomu komerční produkt MS Excel³.

★★★ Formát dat je otevřený

- Přínosy pro uživatele - uživatel při zpracování dat není omezen žádným specifickým nástrojem
- Přínosy/náklady pro vydavatele - nenáročné na publikaci, může však vyžadovat transformaci dat, např. z uzavřeného formátu
- Příkladem může být formát CSV

Teprve v této kategorii se můžeme bavit o „opravdových“ otevřených datech. Resp. data musejí mít stupeň otevřenosti minimálně 3★, aby naplnila základní definici otevřených dat uvedenou výše.

³Toto se netýká formátu .xlsx. Ten již vychází z otevřené specifikace Office Open XML[19]. Data publikovaná v .xlsx formátu tedy můžeme chápat jako 3★.

★★★★ Jednotlivé objekty jsou identifikovány pomocí URI

- Přínosy pro uživatele - uživatel se může na data odkazovat, odkazy si ukládat, případně data snadno kombinovat s jinými (na stejném, nebo vyšším stupni)
- Přínosy/náklady pro vydavatele - náročnější na publikaci
- Příkladem může být formát RDF

Důležité je dbát na to, aby URI nebylo virtuální, resp. aby se po dotázání uživateli vrátil požadovaný obsah. V prostředí WWW je zajištění obsahu typicky praktikováno skrze protokol HTTP.

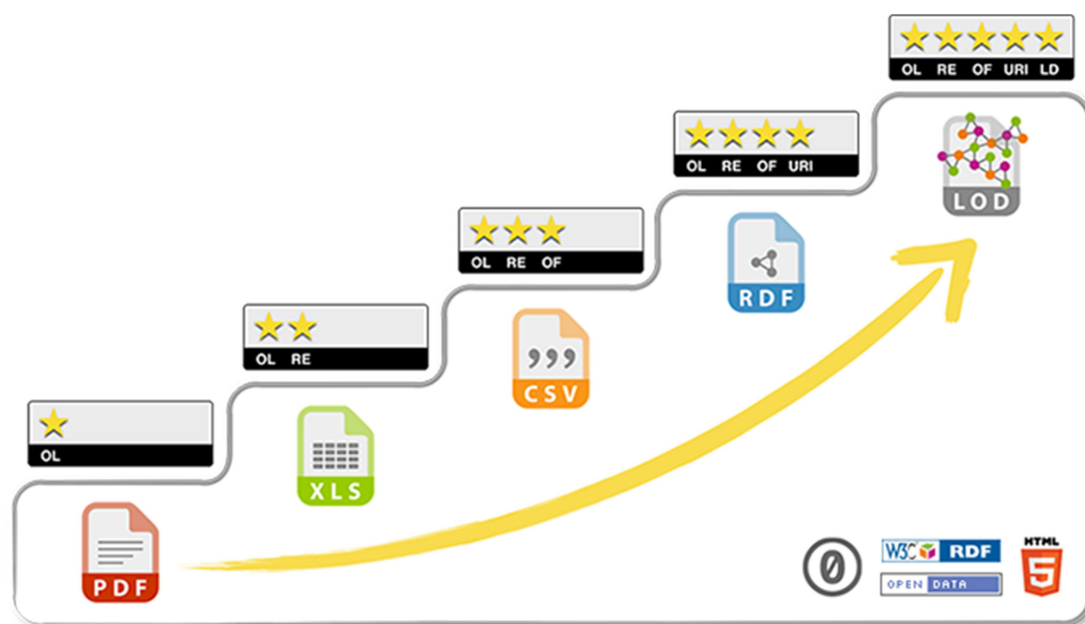
Díky URI identifikaci můžeme data reprezentovat jako orientovaný graf propojených objektů, které na sebe mohou vzájemně odkazovat. K popisu takovýchto dat se používá formát RDF.[2.7]

V prostředí České republiky je tento stupeň považován za nadstandard.

★★★★★ Data jsou propojena se souvisejícími daty

1. Přínosy pro uživatele - vytvoření efektu datové sítě, větší informační hodnota dat
2. Přínosy/náklady pro vydavatele - náročnější na publikaci
3. Příkladem může být formát RDF

V této nejvyšší kategorii se data mohou stát součástí datové sítě propojených grafů.



Obrázek 2.2: Stupně otevřenosti dat, zdroj:[17]

2.4 Propojitelná data (Linked Data)

Linked Data vychází z myšlenky webu aplikované na data. Webu rozumíme jako síti propojených webových stránek. Cílem Linked Data je mít síť propojených, strojově čitelných dat, resp. stavební kámen sémantického webu[20]. Jedná se v podstatě o další krok v evolučním vývoji webu jako takového.

Podle [18] definujeme základní principy Linked Data jako:

1. *Každá entita je identifikována pomocí HTTP URI*
2. *HTTP URI by mělo být vyhledatelné v síti WWW a umožňovat k němu přistupovat a odkazovat se na něj*
3. *Po přístoupení na HTTP URI entity mají být poskytnuty relevantní informace o dané entitě ve standardizovaném formátu či prostřednictvím API⁴*
4. *Data k entitám rozšířit o HTTP URI odkazy na další související entity⁵*

Jak je vidět, Linked Data naplňují všechny požadavky na 5★ kvalitu dat s jednou výjimkou. Linked Data nemusejí být z podstaty otevřenými daty. Určitě si dovedeme představit mnoho scénářů, kdy je přínosem mít propojená, ale privátní data. Typickým příkladem mohou být korporátní intranetové informační systémy. [24]

2.5 Otevřená a propojitelná data (Linked Open Data - LOD)

Otevřená data na úrovni 5★ kvality můžeme tedy chápat jako Linked Open Data. Taková data se mohou stát součástí globálního prostoru sdílených, propojených dat. Připojením datové sady tak můžeme čerpat informační potenciál celého prostoru⁶.

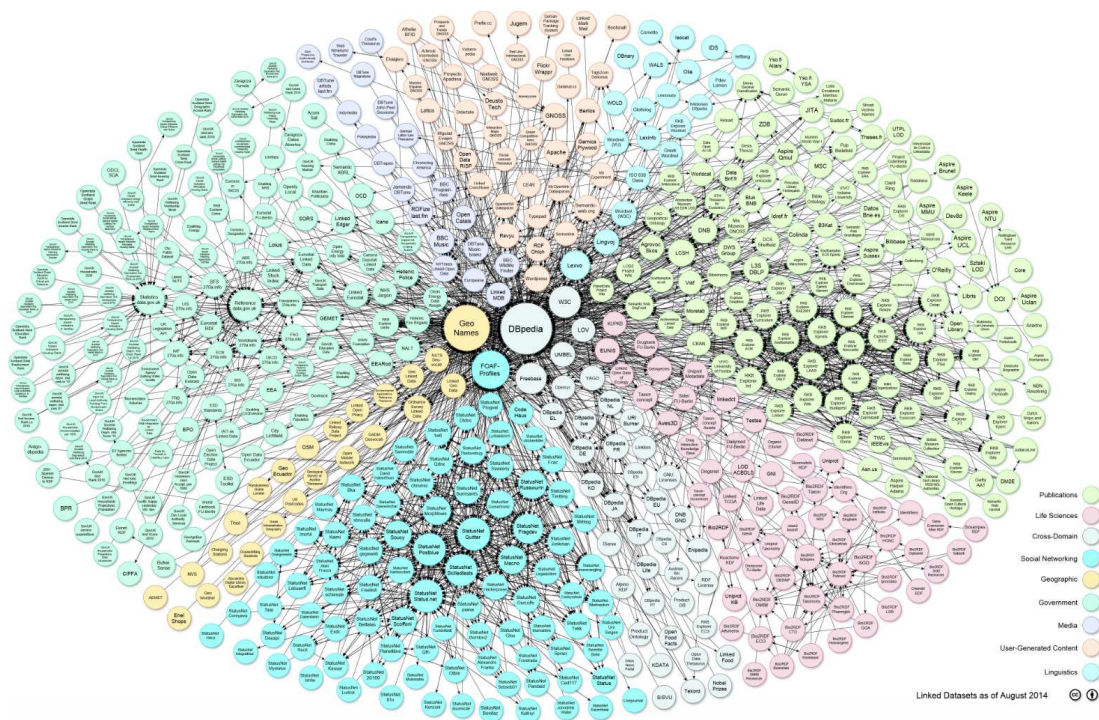
Takový prostor s časem neustále roste. Využití lze nalézt ve většině oblastí lidského konání. Od sdílení a obohacování vědeckých dat, např. biologických, chemických struktur a reakcí s cílem objevů nových postupů v medicíně, přes zpracování dat jednotlivých veřejných správ za účelem kvalitnější veřejné služby až po obohacování kontextu nejrůznějšího mediálního obsahu.

Na obr. 2.3 vidíme příklad vizualizace otevřených a propojených (LOD) dat nazývaný Linked Open Data Cloud. Jedná se o datasety obsahující alespoň 1000 trojic (více v kapitole o RDF) a alespoň 50 odkazů na jiná data ve sdíleném prostoru.

⁴Pro popis Linked Data se typicky používá jazyk RDF[21], k dotazování k datovému API - SPARQL[22]

⁵To nám zaručí, že můžeme procházet jednotlivé entity podobným způsobem jako webové stránky v rámci sítě WWW.

⁶Tvůrci grafu procházejí web a do cloudu přidávají dostupné datasety splňující podmínky Linked Data a podmínky na počet trojic a odkazů. Nezkoumají ale licence jednotlivých datasetů. Některé datasety proto mohou být chráněny specifickými právy.[23]



Obrázek 2.3: Linked Open Data Cloud, Srpen 2014, [23]

2.6 Výhody a přínosy otevřených dat a principů Linked data

Obecné výhody otevřených dat[25, 16]

1. Zapojení uživatelů - kontrola, návrhy ke zlepšení dat
2. Zvýšení transparentnosti vydavatele dat, boj s korupcí
3. Kvalitnější veřejná služba, lepší prezentace subjektu
4. Zvýšení efektivity, úspora nákladů, méně chyb
5. Méně žádostí o data podle zákona č. 106/1999 Sb.[1]
6. Široké možnosti dalšího využití - analýzy, statistiky, vizualizace

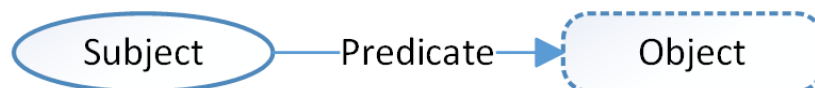
Výhody principů Linked Data[17, 18]

1. Sdílená, rozšiřitelná a snadno znovu použitelná data
2. Data jsou začleněna do kontextu, resp. lze se odkazovat přímo na data
3. Data jsou propojena s dalšími relevantními daty, informační hodnota dat je tedy tím větší, čím více mají vazeb
4. Standardizované formáty pro publikaci

2.7 RDF (Resource Description Framework)

Formát RDF byl vyvinut za účelem snadného strojového zpracování a propojování dat. Jedná se o čistě abstraktní formát udávající, jak data popisovat. Nezabývá se tedy konkrétní podobou výsledných dat.

Základním stavebním kamenem RDF je tvrzení, resp. trojice: **Subjekt - Predikát - Objekt** (viz Obr. 2.4). Subjektem je míněn zdroj, který popisujeme. Predikát je vlastnost, která o objektu něco tvrdí. Objekt je hodnota dané vlastnosti. Jednotlivé trojice mohou na sebe navazovat a vytvořit tak orientovaný graf.



Obrázek 2.4: Základní RDF trojice

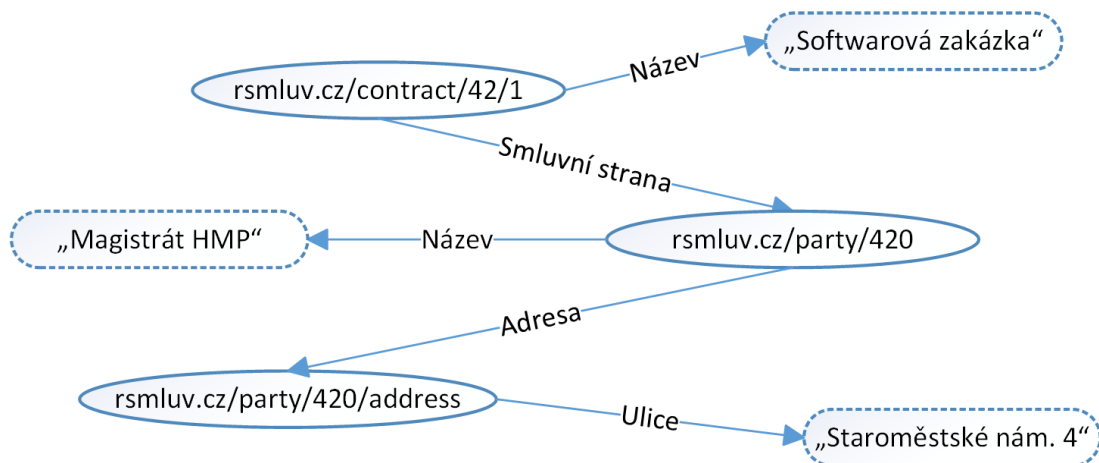
Nyní definujeme několik pravidel a doporučení pro popisování dat v RDF

1. Každý subjekt je jednoznačně identifikován pomocí URI, nebo je označen jako anonymní⁷
2. Objektem je buď hodnota (literál), odkaz na subjekt (resource), nebo je označen jako anonymní
3. Pro každý subjekt je specifikován jeho typ (třída) formou URI
4. Každému predikátu je přiřazen řetězec ve formě URI
5. Jednotlivé URI z bodů 3, 4 by měly odkazovat na konkrétní slovníky tříd a predikátů, resp. ontologie

Na obr. 2.5 vidíme příklad jednoduchého grafu ve formátu RDF (aplikována pravidla 1 a 2). Popisuje 3 subjekty a přiřazuje jim konkrétní vlastnosti. Vidíme, že každý subjekt je identifikován vlastním URI. Díky tomu mohou subjekty na sebe odkazovat. Jednotlivé trojice by pak vypadaly takto:

1. *http://rsmluv.cz/contract/42/1 - Název - Softwarová zakázka*
2. *http://rsmluv.cz/contract/42/1 - Smluví strana - rsmluv.cz/party/420*
3. *http://rsmluv.cz/party/420 - Název - Magistrát HMP*
4. *http://rsmluv.cz/party/420 - Adresa - rsmluv.cz/party/420/address*
5. *http://rsmluv.cz/party/420/address - Ulice - Staroměstské nám. 4*

⁷Subjekty, příp. objekty lze označit jako anonymní, resp. pomocí tzv. Blank node. Na anonymní subjekty, resp. objekty ale nelze přímo přistupovat. Používají se typicky k zapouzdření, či jako kontejnery jiných objektů.

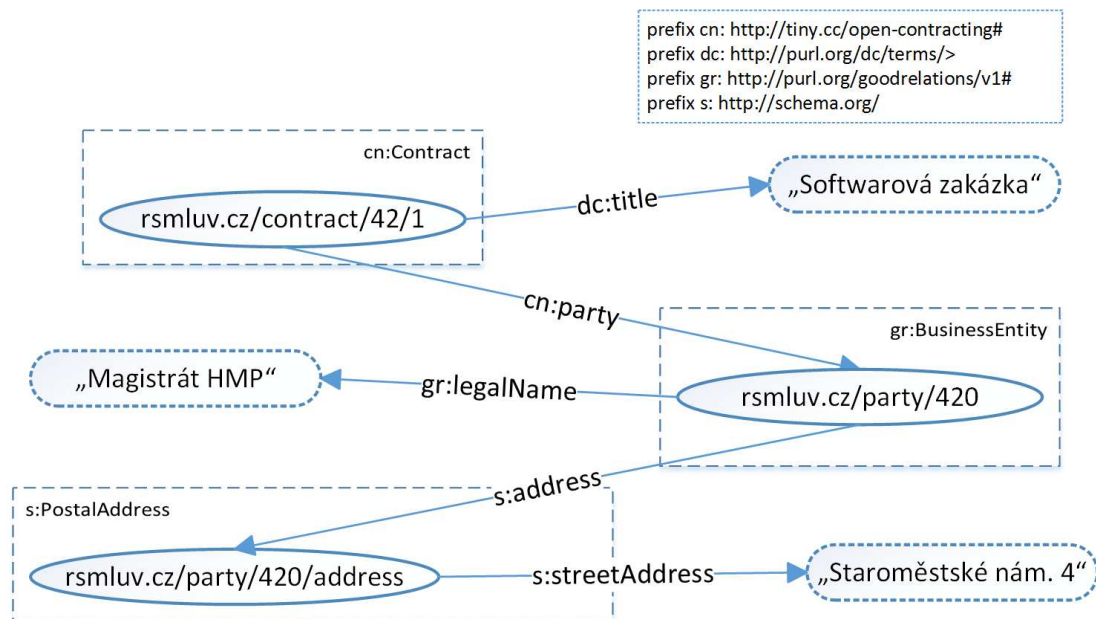


Obrázek 2.5: Jednoduchý RDF graf

Ze zmíněného příkladu ale není zřejmý význam, resp. sémantika jednotlivých subjektů a predikátů. Je tedy důležité jim přiřadit konkrétní typy. Každý typ by měl být popsán v konkrétním slovníku tříd a predikátů. Takovéto slovníky nazýváme ontologiemi. Na obr. 2.6 vidíme zmíněný příklad rozšířený o přiřazené typy (aplikována pravidla 3, 4, 5)⁸. Jednotlivé trojice by pak vypadaly takto:

1. *http://rsmluv.cz/contract/42/1 - rdf:type - cn:Contract*
2. *http://rsmluv.cz/contract/42/1 - dc:title - Softwarová zakázka*
3. *http://rsmluv.cz/contract/42/1 - cn:party - rsmluv.cz/party/420*
4. *http://rsmluv.cz/party/420 - rdf:type - gr:BusinessEntity*
5. *http://rsmluv.cz/party/420 - gr:legalName - Magistrát HMP*
6. *http://rsmluv.cz/party/420 - s:address - rsmluv.cz/party/420/address*
7. *http://rsmluv.cz/party/420/address - rdf:type - s:PostalAddress*
8. *http://rsmluv.cz/party/420/address - s:streetAddress - Staroměstské nám. 4*

⁸Pro zapisování typů se kvůli úspornosti používají prefixy definované typicky na začátku dokumentu.



Obrázek 2.6: RDF graf s přiřazenými typy

2.8 RDF Ontologie

Pod pojmem ontologie si můžeme představit sadu termínů popisujících určitou věcnou oblast. V případě popisování RDF dat definujeme slovník tříd a vlastností (predikátů), které mohou uživatelé ve svých datech používat.

Konkrétní ontologii nelze chápat jako striktně vyžadovaný standard, ale spíše jako sadu doporučení. Buď využijeme k popisu dat nějakou z řady již existujících ontologií, nebo můžeme vytvořit ontologii vlastní. Přesto ale chceme, aby se již existující ontologie používaly co nejvíce. Přínosem je hlavně to, že aplikace a nástroje implementované nad známými ontologiemi budou schopné automaticky rozpoznat naše data⁹.

Základními jazyky pro modelování RDF dat jsou Web Ontology Language (OWL)[30] a RDF Schema (RDFS)[31]. Konkrétní specifikace se provádí opět ve formátu RDF a je publikována pod vlastním URI.

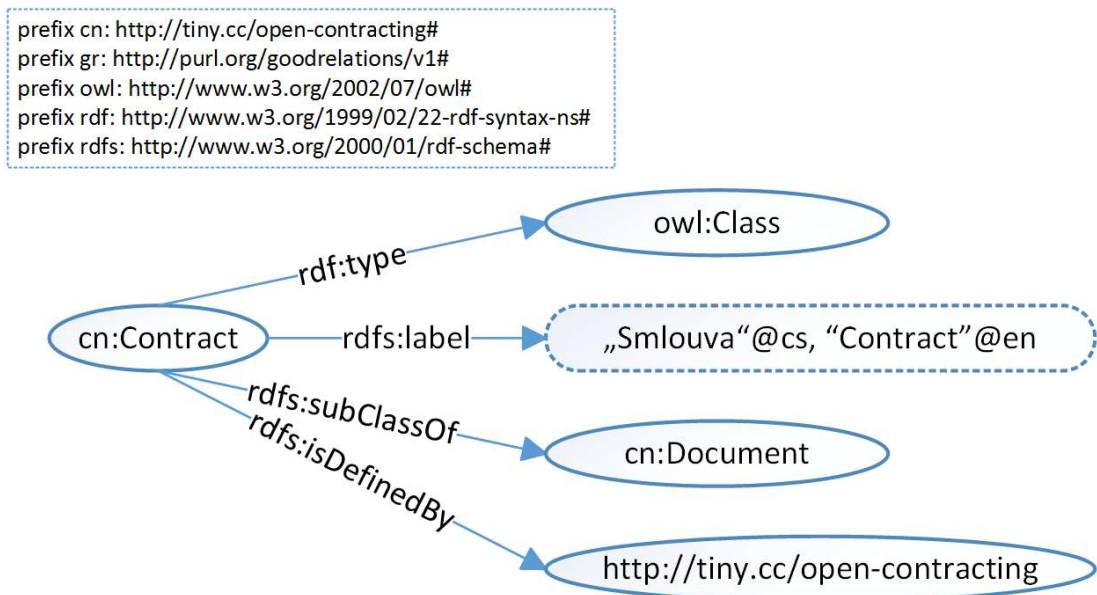
Mezi základní výrazové prostředky jazyka OWL a RDFS patří:

- *owl:Class* - typ entity třída
- *owl:ObjectProperty* - typ entity vlastnost
- *owl:FunctionalProperty* - typ funkcionální vlastnost (může mít nejvýše jednu hodnotu)
- *owl:unionOf* - jeden typ třídy z výčtu musí být vyplněn
- *owl:equivalentClass* - definuje, že se jedná o třídu odpovídající jiné třídě
- *owl:equivalentProperty* - definuje, že se jedná o vlastnost odpovídající jiné vlastnosti

⁹Mezi všeobecně známé ontologie patří např. DublinCore[26], Friend-of-a-Friend[27] nebo Schema[28]. Existuje také katalog ontologií[29]

- *rdfs:label* - popis třídy/vlastnosti
- *rdfs:comment* - komentář třídy/vlastnosti
- *rdfs:domain* - požadovaný typ domény třídy/vlastnosti
- *rdfs:range* - požadovaný rozsah typů třídy/vlastnosti
- *rdfs:isDefinedBy* - definice zdroje třídy/vlastnosti
- *rdfs:subClassOf* - definice, že se jedná o podtřídu určité třídy
- *rdfs:subPropertyOf* - definice, že se jedná o podvlastnost určité vlastnosti

Na obr. 2.7 vidíme příklad části ontologie definující třídu *Contract*. Ontologie nám říká, že se jedná o třídu (typ *owl:Class*) s názvem *Smlouva* (*rdfs:label*), která je podtřídou (*rdfs:subClassOf*) třídy *Document* a je definovaná (*rdfs:isDefinedBy*) v ontologii *http://tiny.cc/open-contracting*. Kdokoli pak bude zpracovávat entitu označenou tímto typem, tak díky přiřazené ontologii bude schopen určit, že se jedná o smlouvu.

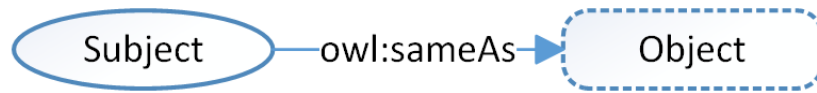


Obrázek 2.7: Ontologie třídy *Contract*

2.8.1 Propojování se souvisejícími entitami

Díky RDF můžeme data reprezentovat jako orientovaný graf. Otázka tedy zní, zdali lze propojovat grafy mezi sebou. Ve formátu RDF je to velmi jednoduché. Jako objekt predikátu stačí položit subjekt z jiného grafu. Díky URI identifikaci entit tedy není rozdílem, zdali je cílovým subjektem entita v lokálních datech, nebo entita cizí.

V rámci propojování dat s jinými datasety však není neobvyklé, že stejné entity jsou reprezentované v různých datasetech pod vlastními URI. Je tedy třeba vyjádřit, že se jedná o data reprezentující stejné entity. V jazyku OWL za tímto účelem existuje predikát *sameAs*, kterým můžeme definovat odpovídající si entity (viz Obr. 2.8).



Obrázek 2.8: Odpovídající si entity

2.9 Publikace

V minulých kapitolách bylo řečeno, jak popisovat data pomocí RDF. Jednalo se o sémantický popis. Pokud však data chceme publikovat, je třeba konkrétního datového formátu, který definuje syntaxi, resp. jak RDF data serializovat. Takových formátů existuje celá řada, např.:

- **N-Triples**[32] - nejjednodušší serializace RDF grafu v podobě výčtu trojic
- **N-Quads**[33] - rozšíření pro N-Triples s možností zaznamenat více grafů
- **RDF/XML**[34] - RDF graf serializovaný do XML, využívající prefixového zápisu
- **Turtle**[35] - úsporný textový formát s možností komprese trojic, využívající prefixových zápisů
- **Trig**[36] - rozšíření Turtle pro použití nad více grafy
- **RDFa**[37] - serializace RDF do (X)HTML dokumentů, využívající prefixového zápisu
- **JSON-LD**[38] - specifický zápis RDF grafu, využívající mapování položek JSON dokumentu na RDF ontologie

Pro potřeby této práce si vystačíme s formáty N-Triples, Turtle a JSON-LD. Vysvětlíme si je na příkladech. Jako data k serializaci použijeme příklad z obr. 2.6.

2.9.1 Příklad dat serializovaných ve formátu N-Triples

Serializace RDF dat do N-Triples je velmi jednoduchá. Jedná se o seznam trojic oddělených tečkou. Každá trojice je uvedena na vlastním řádku. Tento formát nepoužívá prefixové zkracování URI. Je vhodný pro proudové zpracování velkého množství dat (viz Obr. 2.1)¹⁰.

```

1 <http://rsmluv.cz/contract/42/1>
2   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
3   <http://tiny.cc/open-contracting#Contract> .
4 <http://rsmluv.cz/contract/42/1>
5   <http://purl.org/dc/terms/title>
6   "Softwarová zakázka" .
7 <http://rsmluv.cz/contract/42/1>
8   <http://tiny.cc/open-contracting#party>
9   <http://rsmluv.cz/party/420> .

```

¹⁰Trojice nejsou z důvodu přehlednosti uvedeny na samostatných řádcích

```

10
11 <http://rsmluv.cz/party/420>
12   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
13   <http://purl.org/goodrelations/v1#BusinessEntity> .
14 <http://rsmluv.cz/party/420>
15   <http://purl.org/goodrelations/v1#legalName>
16   "Magistrát HMP" .
17 <http://rsmluv.cz/party/420>
18   <http://schema.org/address>
19   <http://rsmluv.cz/party/420/address> .
20
21 <http://rsmluv.cz/party/420/address>
22   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
23   <http://schema.org/PostalAddress> .
24 <http://rsmluv.cz/party/420/address>
25   <http://schema.org/streetAddress>
26   "Staroměstské nám. 4" .

```

Výpis kódu 2.1: Příklad RDF dat - N-Triples

2.9.2 Příklad dat serializovaných ve formátu Turtle

Formát Turtle umožňuje zkracování URI pomocí prefixů. Umožňuje také zkracovat zápis tím, že nemusíme zapisovat opakující se subjekt. Jednotlivé dvojice predikát-hodnota lze tak přehledně mít u jednoho subjektu. Oddělovačem mezi dvojicemi v rámci subjektu je středník, blok informací o daném subjektu je zakončený tečkou. Pro definování typu subjektu se může použít klíčové slovo „a“, namísto predikátu `rdf:type`. Výhodou formátu je úspornost a velmi dobrá lidská čitelnost (viz Kód 2.2).

```

1 @prefix cn: <http://tiny.cc/open-contracting#>.
2 @prefix dc: <http://purl.org/dc/terms/>.
3 @prefix gr: <http://purl.org/goodrelations/v1#>.
4 @prefix s: <http://schema.org/>.
5
6 <http://rsmluv/contract/42/1> a cn:Contract;
7   dc:title "Softwarová zakázka";
8   cn:party <http://rsmluv/party/420>.
9
10 <http://rsmluv/party/420> a gr:BusinessEntity;
11   gr:legalName "Magistrát HMP";
12   s:address <http://rsmluv/party/420/address>.
13
14 <http://rsmluv/party/420/address> a s:PostalAddress;
15   s:streetAddress "Staroměstské nám. 4".

```

Výpis kódu 2.2: Příklad RDF dat - Turtle

Díky dobré čitelnosti, se formát Turtle hojně používá pro zapisování ontologií. V kódu 2.3 vidíme znázorněnou jednoduchou ontologii. Popisuje 2 objekty. Prvním je třída `Contract` (typ `owl:Class`). Definuje, že se jedná o smlouvu, je podtřídou (`rdfs:subClassOf`) třídy `Document` a je definována v ontologii (`rdfs:DefinedBy`) `http://tiny.cc/open-contracting`. Je to serializovaný zápis ontologie z obr. 2.7. Druhým objektem je vlastnost `party` (typ `owl:ObjectProperty`). V predikátu `rdfs:domain` je specifikováno, že vlastnost `party` může být použita u

třech tříd, a to Contract, Order nebo Invoice. Predikát `rdfs:range` znamená, že očekávaný přiřazený objekt je typu `gr:BusinessEntity`.

```
1 @prefix : <http://tiny.cc/open-contracting#> .
2 @prefix dc: <http://purl.org/dc/terms/> .
3 @prefix gr: <http://purl.org/goodrelations/v1#> .
4 @prefix owl: <http://www.w3.org/2002/07/owl#> .
5 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7
8 :Contract a owl:Class ;
9   rdfs:label "Smlouva"@cs, "Contract"@en ;
10  rdfs:subClassOf :Document ;
11  rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
12
13 :party a owl:ObjectProperty ;
14  rdfs:label "Smluvní strana"@cs, "Party"@en ;
15  rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Order
16    :Invoice ) ] ;
17  rdfs:range gr:BusinessEntity ;
18  rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
```

Výpis kódu 2.3: Příklad RDF Ontologie - Turtle

2.9.3 Příklad dat serializovaných ve formátu JSON-LD

JSON-LD je jedním z poměrně nových formátů pro serializaci RDF. Jednou z motivací k vzniku byla snaha využít hojně využívané JSON dokumenty v dnešních aplikacích a co možná nejefektivněji z nich vytvořit RDF data.

Uveďme si modelový příklad. V kódu 2.4 jsou ne-RDF data ve formátu JSON. Jsou validní vůči nějakému JSON Schématu a používají se v konkrétních aplikacích.

V kódu 2.5 máme stejná data v RDF podobě. Jak je vidět, jednotlivým objektům je přiřazen typ a URI. Použije se k tomu klíčových slov `@type`, resp. `@id`. K dokumentu je také přiložen kontext (klíčové slovo `@context`), kde se definuje mapování vlastností původního JSON dokumentu na RDF ontologie. Zachovává se tedy původní struktura JSON dokumentu. Kontext však nemusí být přímo součástí JSON-LD dokumentu, lze se na něj odkazovat.

Výsledkem tedy může být JSON-LD soubor (viz Kód 2.6). Jedná se tedy pouze o lehce rozšířený původní JSON dokument. Z tohoto důvodu bude pravděpodobně takový dokument nadále validní vůči JSON Schématu a použitelný ve stávajících aplikacích. Přináší však tu výhodu, že se zároveň jedná o RDF data.

```

1 {
2   "title": "Softwarová zakázka",
3   "party": {
4
5     "name": "Magistrát HMP",
6     "address": {
7
8       "streetAddress": "Staroměstské nám. 4"
9     }
10  }
11 }

```

Výpis kódu 2.4: Obyčejný JSON dokument

```

1 {
2   "@context": "http://tiny.cc/open-contracting-context",
3
4   "@id": "http://rsmluv.cz/contract/42/1",
5   "@type": "Contract",
6   "title": "Softwarová zakázka",
7   "party": {
8
9     "@id": "http://rsmluv.cz/party/420",
10    "@type": "Party",
11    "name": "Magistrát HMP",
12    "address": {
13
14      "@id": "http://rsmluv.cz/party/420/address",
15      "@type": "Address",
16      "streetAddress": "Staroměstské nám. 4"
17    }
18  }
19 }

```

Výpis kódu 2.5: Příklad RDF dat - JSON-LD

```

1 {
2   "@context": {
3
4     "cn": "http://tiny.cc/open-contracting#",
5     "dc": "http://purl.org/dc/terms/",
6     "gr": "http://purl.org/goodrelations/v1#",
7     "s": "http://schema.org/",
8
9     "Contract": "cn:Contract",
10    "Party": "gr:BusinessEntity",
11    "Address": "s:PostalAddress",
12    "title": "dc:title",
13    "party": "cn:party",
14    "name": "gr:legalName",
15    "address": "s:address",
16    "streetAddress": "s:streetAddress"
17  },
18
19   "@id": "http://rsmluv.cz/contract/42/1",
20   "@type": "Contract",
21   "title": "Softwarová zakázka",
22   "party": {
23

```

```
24  "@id" : "http://rsmluv.cz/party/420" ,
25  "@type" : "Party" ,
26  "name" : "Magistrát HMP" ,
27  "address" : {
28
29      "@id" : "http://rsmluv.cz/party/420/address" ,
30      "@type" : "Address" ,
31      "streetAddress" : "Staroměstské nám. 4"
32  }
33 }
34 }
```

Výpis kódu 2.6: Příklad RDF dat - JSON-LD s Contextem

3. Otevřené smlouvy

3.1 Situace ve veřejné správě ČR

Pokud se veřejná instituce rozhodne pro publikaci údajů o smlouvách, má dnes (rok 2015) v podstatě dvě možnosti. První možností je vyvinutí vlastní iniciativy a zveřejnění smluv na svých webových stránkách. Druhou variantou je využití již existujícího registru smluv na portálu veřejné správy[39]. Registr je to značně minimalistický, ale řešení je to dostačující.

Vzhledem k chystanému zákonu o registru smluv se ale budoucnost stávajícího registru jeví jako značně nejistá. Lze totiž očekávat, že s velkou pravděpodobností vznikne registr zbrusu nový¹.

První otázkou je, kolik veřejných institucí již smlouvy zveřejňuje. Na portálu veřejné správy lze dohledat řádově několik desítek subjektů. O těchto institucích můžeme prohlásit, že oficiálně zveřejňují smlouvy. Informace o subjektech, které zveřejňují na svých webových stránkách, není systematicky zdokumentovaná vůbec. Lze ale očekávat, vzhledem k celkovému množství veřejných institucí a počtu subjektů zveřejňujících na portálu veřejné správy, že se jedná o nepatrný zlomek. Klíčem ke zlepšení situace by mohl být již zmíněný zákon o registru smluv, který mimo jiné ukládá povinnost, že pokud smlouva není zveřejněná na internetu, tak je neplatná.

Další otázkou je, jak mají data o zveřejněných smlouvách vypadat, které položky musí, či nemusí obsahovat. Není přeci cílem, aby každá veřejná instituce zveřejňovala smlouvy jinak. Obecně chybí datový standard a metodika pro zveřejňování smluv. Pokrok v tomto směru udělalo Ministerstvo vnitra ČR, které plánuje vydat sadu standardů pro publikovatelné datové sady veřejných institucí². Bude se mimo jiné jednat o jakési minimální nutné doporučení, co konkrétní datová sada musí obsahovat.

V úvodu již bylo řečeno, že pod záštitou Oživení o.s.[6] a EconLabu[15] (dříve Centrum aplikované ekonomie o.s.) vzniká datový standard pro otevřené smlouvy. Hlavními postavami koordinujícími vývoj standardu se stali PhDr. Ing. Jiří Skuhrovec a Mgr. Lenka Franková. Na tvorbě standardu participují a mohu konstatovat, že základní verze je již hotová³. Velmi pozitivní zprávou je to, že se tento standard s velkou pravděpodobností dostane do oficiálního doporučení Ministerstva vnitra ČR. Zdá se tedy, že celá tato snaha má smysl.

Standardem pro smlouvy to ale nekončí. Myšlenka úzké spolupráce zástupců měst a obcí, akademické a neziskové sféry se osvědčila. Výsledkem je vznik organizace Otevřená města[42], která má za cíl sdružovat veřejné instituce. Pod společnou taktovkou pak financovat společné otevřené projekty. Prvním společným projektem je právě registr smluv. [10, 11, 12, 13]

¹Zákon o Registru smluv - tisk 42[9] byl definitivně schválen 24.11.2015 poslaneckou sněmovnou. Neprošel však ještě celým legislativním procesem. Je už ale téměř jisté, že opravdu vznikne nový registr smluv.

²Standardy publikace a katalogizace otevřených dat veřejné správy ČR[40]

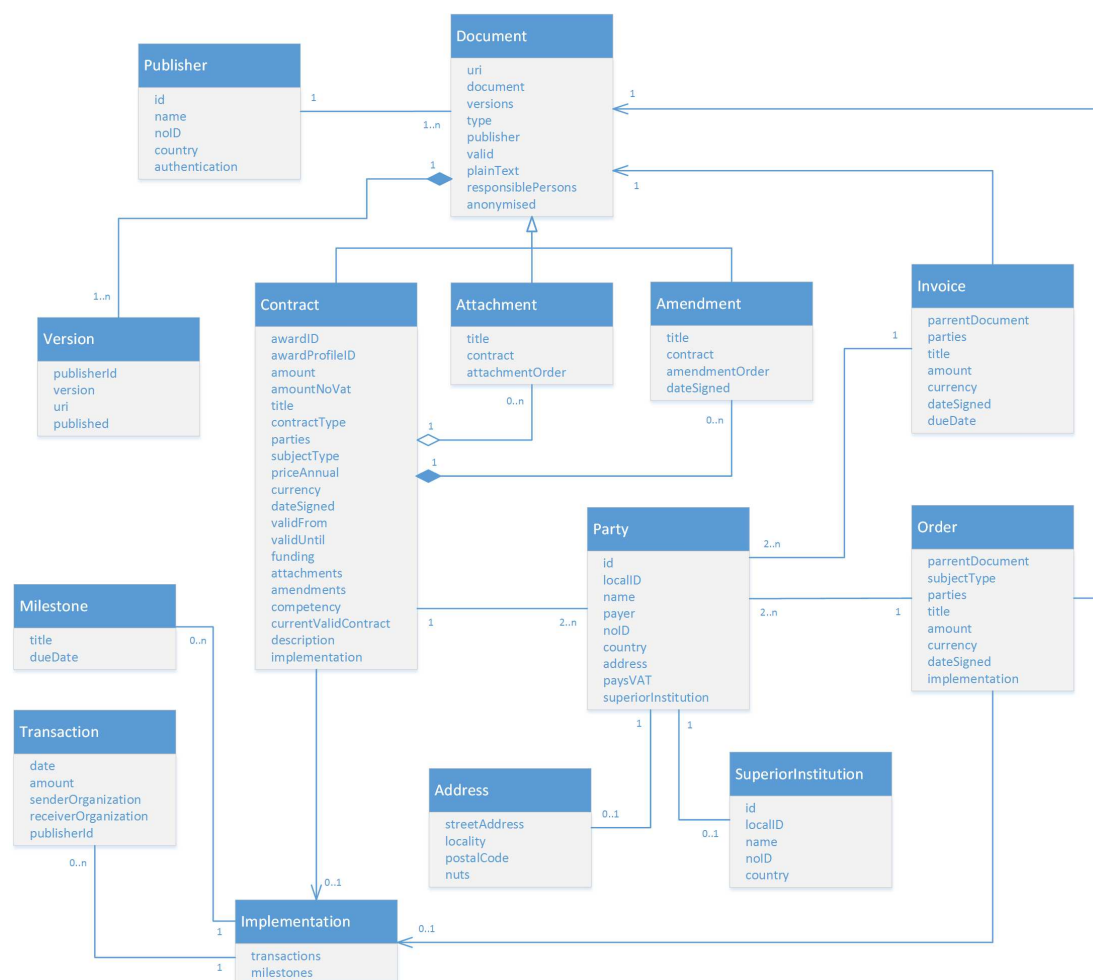
³Původní, nerozšířený koncept standardu vyplynulší z práce akční skupiny je k nalezení na webu iniciativy Bezkorupce[41]

3.2 Standard pro zveřejňování smluv [43, 41]

V této kapitole se podrobněji seznámíme se standardem pro zveřejňování smluv. Nejdříve je vyložena základní struktura datového standardu, poté jsou popsány konkrétní položky standardu a číselníky. Následně jsou popsány způsoby publikace. Na závěr zmíníme několik informací o vznikající metodice pro zveřejňování smluv.

3.2.1 Základní struktura

Základním objektem, který slouží k reprezentaci dat, je dokument. Jedná se o abstraktní entitu, která nabývá tří rozšíření typu smlouva/příloha/dodatek. Tato rozšíření obsahují všechny položky obsažené v dokumentu a navíc konkrétní položky pro daný typ. Smluvní strany jsou separátní objekty navázané buď na smlouvu, objednávku nebo fakturu pomocí jednoznačného identifikátoru. Objednávka a faktura jsou separátní objekty, které se mohou vázat na konkrétní smlouvu/přílohu/dodatek pomocí jednoznačného identifikátoru. Rozšiřující entity mohou být součástí smlouvy, příp. objednávky. Reprezentují důležité události v životním cyklu dokumentu a jednotlivé transakce (viz Obr. 3.1).



Obrázek 3.1: Datový standard pro zveřejňování smluv - UML diagram

Reprezentované entity

- **Dokument** - základní abstraktní struktura pro evidování údajů o smlouvách/přílohách/dodatcích
 - **Smlouva** - detailní popisné údaje smlouvy
 - **Příloha** - popisné údaje přílohy
 - **Dodatek** - popisné údaje dodatku
 - **Vydavatel** - informace o vydavateli, který zveřejňuje údaje o smlouvách
 - **Verze** - identifikace jednotlivé verze dokumentu
- **Smluvní strana** - popisné údaje smluvní strany
 - **Nadřazená instituce** - informace o řídicí nebo ovládací právní osobě vystupující u smluvní strany
 - **Adresa** - podrobné údaje o adrese u smluvní strany
- **Objednávka** - popisné údaje objednávky, jedná se o doplňující informace k smlouvě/příloze/dodatku
- **Faktura** - popisné údaje faktury, jedná se o doplňující informace k smlouvě/příloze/dodatku
- **Rozšiřující entity** - rozšířené informace ke smlouvě, příp. objednávce
 - **Milník** - reprezentuje důležitou událost v životním cyklu smlouvy
 - **Transakce** - reprezentuje proběhlou platbu na základě smlouvy

Datový model je rozdělen do tabulek podle jednotlivých reprezentovaných entit. Každá dílčí položka entity obsahuje tyto informace:

Název pole	Popis
Název pole	Jméno reprezentující danou položku
Datový typ	Přípustný datový typ položky
Validita	Stupeň kvality položky.
Popis	Podrobný popis položky

Tabulka 3.1: Položky tabulek datového standardu, zdroj:[43, 41]

U každé zveřejněné smlouvy rozlišujeme tři stupně validity, resp. správnosti a úplnosti dat: A (kvalitní), B (dobrý), C (základní). Dokumenty musí splňovat alespoň minimální přípustnou kvalitu C. Pokud je nějaký atribut požadován pro stupeň validity C, je níže v textu označen např. takto (C). Položky doplněné systémem jsou označeny (S). Nepovinné položky jsou značeny (N), hvězdička znamená, že položka je kontrolována pokročilejším pravidlem popsáním u konkrétní položky.

Status	Validita	Popis
Nepovinné	N	Nepovinná položka
Základní	C	Povinná položka
Dobrý	B	Rozšiřující položka pro status „Dobrý“
Kvalitní	A	Rozšiřující položka pro status „Kvalitní“
Systémové	S	Položka doplněná systémem

Tabulka 3.2: Validita, zdroj:[43, 41]

Doplňující validační pravidla

Na entity se vztahují další validační pravidla, která nelze přehledně zachytit v rámci popisu jednotlivých položek. Jejich výčet je zde.

- Dokument je buď v strojově čitelném formátu (viz Akceptovatelné soubory), nebo je k němu poskytnut plain text. Pro smlouvy účinné od 1.6.2015⁴ je přípustná pouze varianta ve strojově čitelném formátu.
- U smlouvy typu darovací nesmí být připojeny faktury, ani jedna smluvní strana nesmí být identifikována jako Payer.
- Entita (Vydavatel/Smluvní strana/Nadřazená instituce) má vyplněno buď ID, a nebo NoID = „true“.

Akceptovatelné soubory

Dokumenty připojené ke smlouvám by měly být strojově čitelné, resp. v těchto formátech:

Formát	Validita	Popis
PDF	C	Portable Document Format - ideálně strojově čitelný
DOC	C	Textový dokument Microsoft Word
XLS	C	Tabulka Microsoft Excel
DOCX	B	Textový dokument Microsoft Word
ODT	B	Textový dokument OpenDocument
XLSX	B	Tabulka Microsoft Excel
ODS	B	Tabulka OpenDocument

Tabulka 3.3: Akceptovatelné soubory, zdroj:[43, 41]

⁴Předběžné, bude upřesněno

3.2.2 Reprezentované entity

Dokument

Název pole	Datový typ	Validita	Popis
URI	String URI	S	Jednoznačný identifikátor formou URL. Typicky rsmluv.cz/[Typ]/[Id]/[Version], kde Version je vzestupné číslování verzí při změnách dokumentu či metadat
Document	String URI	S	Adresa URL fyzického umístění dokumentu. Typicky rsmluv.cz/[Typ]/[Id]/[Version]/File, viz akceptovatelné soubory
Versions	Object array	S	Údaje o verzi dokumentu. Viz entita Verze
Type	String/ String enum	C	Typ dokumentu. Nabývá hodnot - Smlouva/Příloha/Dodatek
Publisher	Reference	C	Informace o vydavateli. Viz entita Vydavatel
Valid	Boolean	B/S	Indikuje, zda dokument je platný, tj. nebyl zneplatněn nebo nahrazen novou verzí
PlainText	String	B/S	Prostý text dokumentu (nestrukturovaný, indexovatelný), alternativa pro scanované dokumenty
ResponsiblePersons	String array	B	Výčet odpovědných osob
Anonymised	Boolean	B	Značí, zda-li byla provedena anonymizace dokumentu

Tabulka 3.4: Vlastnosti dokumentu, zdroj:[43, 41]

Vydavatel

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční ID
Name	String	C	Název, případně jméno a příjmení (s tituly)
NoID	Boolean	B	Indikuje že subjekt nemá IČ, nebo zahraniční ID
Country	String	B	Země původu, 3-písmený ISO kód
Authentication	String	S	Značí stupeň ověřenosti zveřejňující strany

Tabulka 3.5: Vlastnosti vydavatele, zdroj:[43, 41]

Verze

Název pole	Datový typ	Validita	Popis
PublisherId	String	N	Libovolný číselný identifikátor verze, spisové číslo apod.
Version	Int	S	Pořadové číslo verze, nejvyšší = aktuální
URI	String URI	S	Identifikátor dané verze
Published	DateTime	S	Datum publikace v systému

Tabulka 3.6: Vlastnosti verze smlouvy, zdroj:[43, 41]

Smlouva

Název pole	Datový typ	Validita	Popis
AwardID	String	N*	Evidenční číslo veřejné zakázky. Uvádí se volitelně, pokud existuje
AwardProfileID	String	N	Číslo zakázky na profilu zadavatele
Amount ⁵	Nullable float	C*	Cena s DPH (u neplátců celková cena). Nejvyšší přípustná hodnota řádného plnění z dané smlouvy, které vynaloží některá smluvní strana. U smluv na dobu určitou se jedná o očekávané celkové finanční plnění strany s nejvyšším plněním, včetně opcí, bez sankcí. U smluv na dobu neurčitou, ve kterých není stanoven strop na celkové plnění, se jedná o nejvyšší očekávané roční plnění. U smluv bez finančního plnění (bartery, darovací smlouvy) je uvedena celková hodnota nefinančního plnění strany s nejvyšším plněním (např. odhadovaná hodnota daru). U smluv s nejasným plněním připustit NULL. Pokud je cena nenulová, tak alespoň jedna Smluvní strana (Party) musí mít příznak Payer = true
AmountNoVat	Nullable float	C*	Cena bez dph, uvádí se povinně pouze v případě, že Amount je s DPH
Title	String	C	Předmět smlouvy
ContractType	String	C	Číselník typů smlouvy, viz Číselníky
Parties	StringURI/ Int array	C	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entita Smluvní strana
SubjectType	String	B	Číselník typů zboží/slужeb, viz Číselníky
PriceAnnual	Boolean	B	Identifikuje, pokud je v Amount roční částka
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát

⁵U položek Amount a AmountNoVat připustíme místo ceny vyplněný objekt složený z položek AmountValue (cena) a Currency (měna). Je to z důvodu lepšího zapouzdření informací o ceně.

Název pole	Datový typ	Validita	Popis
DateSigned	Date	B	Datum posledního podpisu
ValidFrom	Date	B	Datum účinnosti smlouvy
ValidUntil	Date	B	Datum ukončení účinnosti smlouvy (poslední plnění), NULL pro smlouvy na dobu
Funding	String	B	Převažující financování – vlastní, případně název dotačního titulu (bude kontrolován proti číselníku, viz Číselníky)
Attachments	String URI-array	B	Seznam URI identifikátorů příloh. Viz entita Příloha
Amendments	String URI-array	B	Seznam URI identifikátorů dodatků. Viz entita Dodatek
Competency	String/String enum	A	Indikuje, zda-li se jedná o soukromoprávní nebo veřejnoprávní smlouvu
CurrentValidContract	String URI	A	Aktuálně platné znění smlouvy (se zapracovanými dodatky)
Description	String	A	Popis předmětu smlouvy
Implementation	Object	A	Objekt reprezentující transakce a milníky, viz entita Implementation

Tabulka 3.7: Vlastnosti smlouvy, zdroj:[43, 41]

Příloha

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
Contract	String URI	C	Jednoznační identifikátor smlouvy
AttachmentOrder	Int	B	Pořadové číslo přílohy

Tabulka 3.8: Vlastnosti přílohy, zdroj:[43, 41]

Dodatek

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
Contract	String URI	C	Jednoznační identifikátor smlouvy
AmendmentOrder	Int	B	Pořadové číslo dodatku (podle času podpisu)
DateSigned	Date	B	Datum podpisu

Tabulka 3.9: Vlastnosti dodatku, zdroj:[43, 41]

Smluvní strana

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční id
LocalID	String URI/Int	C	Jednoznačný identifikátor v rámci dokumentu
Name	String	C	Název, případně jméno a příjmení (s tituly)
Payer	Boolean	C*	Identifikuje stranu která bude finančně plnit, pokud není zřejmé, nevyplňuje se
NoID	Boolean	B	Indikuje že subjekt nemá IČ, nebo zahraniční ID
Country	String	B	Země původu, 3-písmený ISO kód
Address	String/Referer	A	Adresa subjektu, případně "Anonymizováno". Umožňuje zadat adresu jako prostý řetězec, nebo strukturovaně, viz entitia Adresa
PaysVAT	Boolean	A	Indikuje, zda-li je subjekt plátce DPH
SuperiorInstitution	Reference	N/S	Řídící nebo ovládající právnická osoba, v případě veřejnoprávních smluv nadřazený správní orgán. Viz Nadřazená instituce

Tabulka 3.10: Vlastnosti smluvní strany, zdroj:[43, 41]

Nadřazené instituce

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční id
LocalID	String URI/Int	C	Jednoznačný identifikátor v rámci dokumentu
Name	String	C	Název, případně jméno a příjmení (s tituly)
NoID	Boolean	B	Indikuje že subjekt nemá IČ, nebo zahraniční ID
Country	String	B	Země původu, 3-písmený ISO kód

Tabulka 3.11: Vlastnosti nadřazené instituce, zdroj:[43, 41]

Adresa

Název pole	Datový typ	Validita	Popis
StreetAddress	String	A	Ulice, případně "Anonymizováno"
Locality	String	A	Město, případně "Anonymizováno"
PostalCode	Integer	A	PSC, případně "Anonymizováno"
Nuts	String	A	Normalizovaná klasifikace územních celků (např. Praha - CZ010), případně "Anonymizováno"

Tabulka 3.12: Vlastnosti adresy, zdroj:[43, 41]

Objednávka

Název pole	Datový typ	Validita	Popis
ParrentDocument	String URI	N	Jednoznačný identifikátor dokumentu
SubjectType	String	N	Číselník typů zboží/služeb, viz Číselníky
Parties	String URI/Int array	N	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entita Smluvní strana
Title	String	C	Předmět
Amount	Float	C	Cena s DPH
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát
DateSigned	Date	B	Datum posledního podpisu
Implementation	Object	A	Objekt reprezentující transakce a milníky, viz entita Implementation

Tabulka 3.13: Vlastnosti objednávky, zdroj:[43, 41]

Faktura

Název pole	Datový typ	Validita	Popis
ParrentDocument	String URI	N	Jednoznačný identifikátor dokumentu
Parties	String URI/Int array	N	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entita Smluvní strana
Title	String	C	Předmět
Amount	Float	C*	Cena s DPH (u neplátců celková cena).
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát
DateSigned	Date	B	Datum posledního podpisu
DueDate	Date	B	Datum splatnosti

Tabulka 3.14: Vlastnosti faktury, zdroj:[43, 41]

Rozšiřující entity

Implementace

Název pole	Datový typ	Validita	Popis
Milestones	Object arra	A	Milníky, pro volnou evidenci událostí (obnova smlouvy, předání apod.). Viz entita Milník
Transactions	Object array	A	Seznam transakcí, tedy proběhlých plateb na základě smlouvy. Viz entita Transakce

Tabulka 3.15: Vlastnosti implementace, zdroj:[43, 41]

Milník

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
DueDate	String	C	Datum

Tabulka 3.16: Vlastnosti milníku, zdroj:[43, 41]

Transakce

Název pole	Datový typ	Validita	Popis
Date	DateTime	C	Datum a čas proběhlé transakce
Ammount	Float	C	Zaplacená cena s DPH, vždy stejná měna jako v Currency
SenderOrganization	Reference	C	Informace o odesílateli. Viz entita Party
ReceiverOrganization	Reference	C	Informace o příjemci. Viz entita Party
PublisherId	String	B	Libovolný číselný identifikátor transakce, unikátní v rámci smlouvy

Tabulka 3.17: Vlastnosti transakce, zdroj:[43, 41]

3.2.3 Číselníky

V následujících tabulkách 3.18,3.19 jsou znázorněny přípustné hodnoty číselníků Typ dokumentu (vlastnost Type u entity Dokument) a Typ smlouvy (vlastnost ContractType u entity Smlouva). Číselník Typ zboží a služeb (položka Subject-Type u entity Smlouva) je zveřejněn na portálu informačního systému o veřejných zakázkách⁶.

Hodnota
Smlouva
Příloha
Dodatek

Tabulka 3.18: Číselník typu dokumentu, zdroj:[43, 41]

Hodnota
Nájemní smlouva
Darovací smlouva
Kupní smlouva
Směnná smlouva
Pojistná smlouva
Smlouva o výpůjčce
Licenční smlouva
Mandátní smlouva
Leasingová smlouva
Pachtovní smlouva
Smlouva o zřízení věcného břemene
Smlouva o provedení stavby
Smlouva o provedení práce
Smlouva o provedení uměleckého výkonu
Smlouva o úvěru
Smlouva o uzavření budoucí smlouvy
Veřejnoprávní smlouva
Jiná

Tabulka 3.19: Číselník typu smlouvy, zdroj:[43, 41]

⁶Dostupné na portálu informačního systému o veřejných zakázkách[44]. Konkrétní vymezení přípustných hodnot ještě není specifikováno.

3.3 Publikace

Pro potřeby publikace je třeba zvolit vhodný datový formát v kterém budou otevřené smlouvy přenositelné. Jako kritéria výběru vhodného formátu stanovíme čtyři podmínky:

- otevřený datový formát - tím zaručíme otevřená data na úrovni kvality 3★
- obecná znalost a jednoduchost datového formátu - cílem je, aby valná většina IT specialistů ve veřejných institucích formát znala
- existence volně dostupných nástrojů k čtení a zpracování datového formátu
- možnost tvorby datového schématu - resp. možnost určit soustavu specifikací a pravidel, jak má datový soubor vypadat, aby byl validní

Není překvapující, že obecně nejznámějšími datovými formáty splňujícími výše zmíněná pravidla jsou formáty XML (Extensible Markup Language) a JSON (JavaScript Object Notation)[45]. Vzhledem k úspornosti a možností rychlejšího zpracování padla volba na formát JSON.

Pokud však chceme, aby datový standard byl součástí plánovaného doporučení Ministerstva vnitra ČR, tak je nutné podporovat také formát CSV (Comma-separated values)[46]. Jedná se o jednoduchý, otevřený datový formát, ale s plochou strukturou. Publikace smluv v CSV si tedy vyžádá řadu omezení. [47, 16]

3.3.1 JSON

Základní strukturu datového souboru lze vidět z tabulky 3.20. Položky Id, Date a Language slouží k popisu datového souboru jako celku. Položky Documents, Parties, Orders a Invoices už obsahují konkrétní výčty entit ze standardu. Položky vyznačené stupněm validity C jsou povinné.

Ke konkrétní specifikaci jednotlivých položek ve formátu JSON se používá JSON Schema[48]⁷. Lze v něm definovat konkrétní elementy a podelementy, výchozí hodnoty, datové typy, požadovaný obsah apod. Příklad JSON Schématu vycházejícího z datového standardu lze nalézt na přiloženém datovém nosiči⁸.

Datový soubor, validní vůči JSON schématu, s jednou smlouvou a dvěma smluvními stranami můžeme vidět na příkladu kódu 3.1.

⁷Popisem způsobu zápisu konkrétních položek se v rámci této práce zabývat nebudeme

⁸Nebo online na GitHubu[49].

Název pole	Datový typ	Validita	Popis
Id	String	C	Jednoznačný identifikátor souboru
Date	DateTime	C	Datum publikace souboru
Documents	Object array	C	Seznam jednotlivých smluv/příloh/dodatků
Language	String	C	Specifikace jazyka pro data. Doporučuje se použití dvou znakového ISO 639-1
Parties	Object array	N	Výčet smluvních stran
Orders	Object array	N	Seznam objednávek
Invoices	Object array	N	Seznam faktur

Tabulka 3.20: Struktura datového souboru, zdroj:[43, 41]

```

1 {
2   "id": "89f689cd-e784-4374-bb17-94144679d46f",
3   "published": "2014-03-25T23:20:50+01:00",
4   "language": "cs",
5
6   "documents": [
7     {
8       "uri": "http://rsmluv.cz/smlouva/12345",
9       "document": "http://rsmluv.cz/smlouva/12345/Smlouva12345.docx",
10      },
11     "type": "Smlouva",
12     "valid": true,
13     "anonymised": false,
14
15     "awardID": "486026",
16     "awardProfileID": "OI-010143",
17     "amount": 584520.00,
18     "title": "Brno, Vackova, Šafaříkova – rekonstrukce kanalizace
19     a vodovodu",
20     "contractType": "Kupní smlouva",
21     "subjectType": "Právní, finanční překladatelské, pojišťovnické
22     , poradenské a jiné služby",
23     "priceAnnual": false,
24     "currency": "CZK",
25     "dateSigned": "2011-11-16",
26     "validFrom": "2011-11-02",
27     "validUntil": "2012-06-30",
28     "funding": "vlastní",
29     "competency": [ "Soukromoprávní smlouva" ],
30     "currentValidContract": "http://zakazky.bрно.cz/?pg=detail&id=18249&list=135",
31     "description": "Projektová dokumentace pro stavební povolení a
32     zadání stavby bude řešit rekonstrukci stávající kanalizační
33     stoky z profilu DN 500 na DN 800/1200 v délce 146 m, rekonstrukci
34     kanalizačních přípojek pod veřejným prostranstvím a přepojení vš
35     ech stávajících dešťových vpustí a také vybourání vozovek a chodn
36     íků nad rýhou a zásyp rýhy recyklátem. Součástí bude inženýrsko-
37     geologický průzkum, geodetické zaměření dotčené oblasti,

```

inventarizace zeleně, výkaz výměr, položkový rozpočet a výkon autorského dozoru až do dokončení stavby. Dokumentace bude projednána s orgány státní správy a s účastníky stavebního řízení a jejich připomínky budou do dokumentace zapracovány.”,

```
29
30   "responsiblePersons" : [ "Ing. Petr Vokřál ", "Mgr. Adriana
    Krnáčová, MBA" ],
```

```
31
32   "publisher" : {
33     "id" : "6003508",
34     "name" : "Statutární město Brno",
35     "noID" : false,
36     "country" : "CZE",
37     "authentication" : "email"
38   },
39
40   "parties" : [ 132456, 987654 ],
41
42   "Implementation" : {
43     "milestones" : [
44       {
45         "title" : "Výpověď smlouvy",
46         "dueDate" : "2012-06-20T23:20:50+01:00"
47       }
48     ],
49     "transactions" : [
50       {
51         "publisherId" : "1269483",
52         "date" : "2012-01-01T18:35:20+01:00",
53         "amount" : 300000,
54         "senderOrganization" : 987654,
55         "receiverOrganization" : 132456
56       },
57       {
58         "publisherId" : "934584",
59         "date" : "2012-02-01T09:13:40+01:00",
60         "amount" : 284520,
61         "senderOrganization" : 987654,
62         "receiverOrganization" : 132456
63       }
64     ]
65   },
```

```
66
67   "versions" : [
68     {
69       "version" : 1,
70       "uri" : "http://rsmluv.cz/smlouva/12345/verze/1",
71       "published" : "2014-09-15T23:20:50+01:00"
72     },
73     {
74       "version" : 2,
75       "uri" : "http://rsmluv.cz/smlouva/12345/verze/2",
76       "published" : "2015-03-15T14:35:28+01:00"
77     }
78   ]
79 }
80 ],
```

```

82  "parties" : [
83    {
84      "id" : "44992785",
85      "localID" : 132456,
86      "name" : "Statutární město Brno",
87      "payer" : false ,
88      "noID" : false ,
89      "country" : "CZE" ,
90      "address" : {
91        "streetAddress" : "Dominikánské náměstí 196/1" ,
92        "locality" : "Brno-město , Brno" ,
93        "postalCode" : 60200,
94        "nuts" : "CZ064"
95      } ,
96      "superiorInstitution" : {
97        "id" : "00064581" ,
98        "localID" : 56486,
99        "name" : "Magistrát hlavního města Prahy" ,
100       "noID" : false ,
101       "country" : "CZE"
102     }
103   } ,
104   {
105     "id" : "46347011" ,
106     "localID" : 987654,
107     "name" : "Kovoprojekta Brno a.s." ,
108     "payer" : true ,
109     "noID" : false ,
110     "country" : "CZE" ,
111     "address" : {
112       "streetAddress" : "Šumavská 416/15" ,
113       "locality" : "Ponava , Brno" ,
114       "postalCode" : 60200,
115       "nuts" : "CZ064"
116     }
117   }
118 ]
119 }

```

Výpis kódu 3.1: JSON soubor s jednou smlouvou

3.3.2 CSV

Formát CSV je jednoduchou plochou strukturou, nelze tedy pomocí tohoto formátu zaznamenat úplnou strukturu datového standardu. Řešením by mohlo být rozdělit údaje o smlouvách do sady CSV souborů. Tím se ale ztrácí výhoda jednoduchosti CSV. Cílem publikace v CSV je maximální jednoduchost pro vydavatele. Proto jsme přistoupili k následujícím omezením (seznam položek lze nalézt v tabulce 3.21):

- Vše je smlouvou, tedy nebudeme evidovat dodatky, přílohy, faktury a objednávky
- Validita je omezena pouze na povinné (červeně v tabulce 3.21) a nepovinné položky
- Vypuštěny/omezeny vlastnosti u smlouvy
 - URI - nahrazeno odkazem na podrobné údaje o smlouvě
 - Type
 - Verzování, resp. vypuštěna vazba na verze a vlastnost Valid
 - PlainText
 - Vydavatel, převážně proto, že MV má pro vydavatele speciální strukturu
 - Pouze jedna zodpovědná osoba
 - Vypuštěny rozšiřující entity - milníky a transakce
- Umožněny pouze dvě smluvní strany - Publisher a Partner, a to s vlastnostmi
 - ID, Name, Country a Address
- Nové vlastnosti
 - LocalID - Libovolný číselný identifikátor smlouvy, spisové číslo apod.
 - NumberOfAmendments - Počet dodatků
 - LastAmendmentDateSigned - Datum podpisu posledního dodatku
 - FirstInvoiceDueDate - Datum splatnosti první faktury
 - LastInvoiceDueDate - Datum splatnosti poslední faktury
 - TotalFillingValue - Celkový objem plnění
 - Payer - Publisher / Partner - identifikuje stranu, která má poskytovat vyčíslené finanční plnění (tedy je typicky odběratelem zboží / služeb)

Datový standard pro formát CSV

Název pole	Datový typ	Popis
Amount	Nullable float	Cena s DPH. Nejvyšší přípustná hodnota řádného plnění z dané smlouvy, které vynaloží některá smluvní strana. U smluv na dobu určitou se jedná o očekávané celkové finanční plnění strany s nejvyšším plněním, včetně opcí, bez sankcí. U smluv na dobu neurčitou, ve kterých není stanoven strop na celkové plnění, se jedná o nejvyšší očekávané roční plnění. U smluv bez finančního plnění (bartery, darovací smlouvy) je uvedena celková hodnota nefinančního plnění strany s nejvyšším plněním (např. odhadovaná hodnota daru). U smluv s nejasným plněním připustit NULL.
AmountNoVat	Nullable float	Cena bez DPH (u neplátců celková cena). Nejvyšší přípustná hodnota řádného plnění z dané smlouvy, které vynaloží některá smluvní strana. U smluv na dobu určitou se jedná o očekávané celkové finanční plnění strany s nejvyšším plněním, včetně opcí, bez sankcí. U smluv na dobu neurčitou, ve kterých není stanoven strop na celkové plnění, se jedná o nejvyšší očekávané roční plnění. U smluv bez finančního plnění (bartery, darovací smlouvy) je uvedena celková hodnota nefinančního plnění strany s nejvyšším plněním (např. odhadovaná hodnota daru). U smluv s nejasným plněním připustit NULL.
Title	String	Předmět smlouvy
ContractType	String	Číselník typů smlouvy dle http://standard.zindex.cz/doku.php/cs/-standard/codelists
Currency	String	Měna, 3-písmenný, ISO 4217 formát
PublisherName	String	Název, případně jméno a příjmení (s tituly)
PartnerName	String	Název, případně jméno a příjmení (s tituly)
SubjectType	String	Převažující Typ zboží/sluzeb podle cpv číselníku, viz http://standard.zindex.cz/doku.php/cs/-standard/codelists
PriceAnnual	Boolean	Identifikuje, pokud je v Amount roční částka (přípustné jen u smluv na dobu neurčitou)
DateSigned	Date	Datum posledního podpisu
ValidFrom	Date	Datum účinnosti smlouvy
ValidUntil	Date	Datum ukončení účinnosti smlouvy (poslední plnění), NULL pro smlouvy na dobu neurčitou

Název pole	Datový typ	Popis
Funding	String	Převažující financování – vlastní, případně název dotačního titulu (bude kontrolován proti číselníku)
ResponsiblePerson	String	Odpovědná osoba (např. jméno příkazce operace)
Anonymised	Boolean	Značí, zda-li byla provedena anonymizace dokumentu
PublisherCountry	String	Země původu, 3-písmený ISO kód
PartnerCountry	String	Země původu, 3-písmený ISO kód
NumberOfAmendments	Int	Počet dodatků
LocalID	String	Libovolný identifikátor smlouvy zveřejňujícího subjektu, spisové číslo apod
URI	String URI	Odkaz na stránku s více informacemi o smlouvě
Document	String URI	Odkaz na soubor dokumentu (pokud možno včetně příloh)
AwardID	String	Evidenční číslo veřejné zakázky, pokud existuje
AwardProfileID	String	Číslo zakázky na profilu zadavatele, pokud existuje
Competency	String	Indikuje, zda-li se jedná o soukromoprávní nebo veřejnoprávní smlouvu
CurrentValidContract	String URI	Aktuálně platné znění smlouvy (se zapracovanými dodatky)
Description	String	Popis předmětu smlouvy
PublisherID	String	Identifikační číslo osoby, v ČR IČO, lze vložit i zahraniční ID
PublisherAdress	String	Adresa subjektu, případně "Anonymizováno"
PartnerID	String	Identifikační číslo osoby, v ČR IČO, lze vložit i zahraniční ID
PartnerAdress	String	Adresa subjektu, případně "Anonymizováno"
LastAmendmentDateSigned	Date	Datum podpisu posledního dodatku
FirstPaymentDueDate	Date	Datum splatnosti první platby
LastPaymentDueDate	Date	Datum splatnosti poslední platby
TotalPaidVolume	Float	Celkový objem plnění
Payer	String	Publisher / Partner - identifikuje stranu, která má poskytovat vyčíslené finanční plnění (tedy je typicky odběratelem zboží / služeb)

Tabulka 3.21: Datový standard serializovaný do CSV, zdroj:[43, 41]

3.4 Metodika zveřejňování smluv

Spolu s datovým standardem vzniká i metodika mající za cíl technicky i věcně datový standard popsat. Tvorbu této metodiky jsem pod taktovkou Jiřího Skuhrovce dostal na starost. Obsah této metodiky vznikal souběžně s touto prací.

Jedná se o jednoduchou webovou aplikaci na bázi wikipedie. Implementována je pomocí nástroje Dokuwiki[50]. Dostupná je pod hlavičkou EkonLabu[43]⁹

HISTORIE: CS / CONTRACT

Metodika zveřejňování smluv [Index](#) Přihlášen(a) jako: Pavel Hryzlik (pavel.hryzlik)

Dokument

Dokument je základní strukturou pro evidování údajů o smlouvách/přílohách/dodatcích. Každý dokument má přiděleno jednoznačné ID a typ, zda-li se jedná o smlouvu, přílohu nebo dodatek. Pro jednotlivé typy rozlišujeme různé datové struktury.

Název pole	Datový typ	Validita	Popis
URI	String URI	S	Jednoznačný identifikátor formou URL. Typicky <code>rsmulov.cz/[Typ]/[Id]/[Version]</code> , kde Version je vzestupně číslovaní verzi při změnách dokumentu či metadat
Document	String URI	S	Adresa URL fyzického umístění dokumentu. Typicky <code>rsmulov.cz/[Typ]/[Id]/[Version]/File</code> , akceptovatelné soubory viz. Validita
Versions	Object array	S	Údaje o verzí dokumentu. Viz. Verze
Type	String/String enum	C	Typ dokumentu. Nabývá hodnot - Smlouva/Příloha/Dodatek
Publisher	Reference	C	Informace o vydavateli. Viz. Vydavatel
Valid	Boolean	B/S	Indikuje, zda dokument je platný, tj. nebyl zneplatněn nebo nahrazen novou verzí
PlainText	String	B/S	Prostý text dokumentu (nestrukturovaný, indexovatelný), alternativa pro scanované dokumenty
ResponsiblePersons	String array	B	Výčet odpovědných osob
Anonymised	Boolean	B	Značí, zda-li byla provedena anonymizace dokumentu

[Upravit](#)

Vydavatel

Informace o vydavateli, který zveřejňuje údaje o smlouvách.

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční ID
Name	String	C	Název, případně jméno a příjmení (s tituly)
NoID	Boolean	B	Indikuje že subjekt nemá IČ, nebo zahraniční ID
Country	String	B	Země původu, 3-písmenný ISO kód
Authentication	String	S	Značí stupeň ověření zveřejňující strany

[Upravit](#)

Verze

Identifikace jednotlivé verze dokumentu.

Název pole	Datový typ	Validita	Popis
Publisherid	String	N	Libovolný číselný identifikátor verze, spisové číslo apod.
Version	Int	S	Pořadové číslo verze, nejvyšší = aktuální
URI	String URI	S	Identifikátor dané verze
Published	DateTime	S	Datum publikace v systému

[Upravit](#)

cs/standard/schema/document.txt · Poslední úprava: 2015/11/21 14:22 autor: pavel.hryzlik

Obrázek 3.2: Metodika zveřejňování smluv, zdroj:[43]

⁹Inspirováno standardem Open Data Contracting[51]

4. Otevřené smlouvy jako Linked Data

V minulé kapitole [3] bylo řečeno, jak prezentovat smlouvy jako otevřená data. Data serializovaná do formátu JSON, či CSV můžeme kategorizovat stupněm 3★. Pokud však chceme dosáhnout otevřenosti dat kategorie 5★, je třeba provést několik dalších kroků:

- Identifikovat reprezentované objekty a vlastnosti pomocí URI
- Vytvořit strojově srozumitelné struktury, resp. napojit data na konkrétní slovníky tříd a predikátů - ontologie
- Propojit smlouvy pomocí odkazů na související data

4.1 Přiřazení identifikátorů jednotlivým entitám otevřených smluv

K jednoznačné identifikaci každé entity nám stačí její *typ* a *Id*. Výjimku tvoří dokumenty, které jsou verzované. K nim je nutné přidat informaci o konkrétní verzi. Dále chceme vyjádřit vztah podřízené entity k nadřízené. Řešením je opět přidání informací o typu podřízené entity, příp. jejího *Id*. Resp. základní URI schéma bude:

http://[domain]/[type]/[entityId]/[versionId]/[childEntity]/[childEntityId]

- *domain* je doména instituce publikující smlouvy
- *type* značí typ reprezentované entity
- *entityId* je jednoznačný identifikátor entity
- *versionId* určuje konkrétní verzi entity (pokud je entita verzovaná)
- některé entity mohou mít i podelementy reprezentované pomocí *childEntity*, resp. identifikátorem *childEntityId*

Výsledné identifikátory entit:

- **Document** (Dokument) - *http://[domain]/[type]/[documentId]/[versionId]*
 - Type může nabývat hodnot *contract/attachment/amendment* - resp. jedná se v podstatě o hodnotu položky *Uri* z datového standardu
- **Publisher** (Vydavatel) - *http://[domain]/publisher*
 - Jedná se o podřízenou položku dokumentu. Předpokládáme, že instituce publikující na konkrétní doméně odpovídá konkrétnímu vydavateli smluv. Je vhodné, aby každý vydavatel publikoval pod unikátní doménou.

- **Version** (Verze) - *http://[domain][type]/[documentId]/[versionId]/version*
 - Jedná se o podřízenou položku dokumentu
- **Order** (Objednávka) - *http://[domain]/order/[orderId]*
- **Invoice** (Faktura) - *http://[domain]/invoice/[invoiceId]*
- **Implementation** (Implementace)
 - *http://[domain]/[type]/[documentId]/[versionId]/implementation*
 - * Pokud se jedná o podřízenou položku dokumentu
 - *http://[domain]/order/[orderId]/implementation*
 - * Pokud se jedná o podřízenou položku objednávky
- **Milestone** (Milník)
 - Zde si dovolíme drobné zjednodušení, a to vynechání implementation z identifikátoru. Adresa bude jednodušší, informační hodnota však zůstane stejná
 - *http://[domain]/[type]/[id]/[versionId]/milestone/[milestoneId]*
 - * Milník u dokumentu
 - *http://[domain]/order/[orderId]/milestone/[milestoneId]*
 - * Milník u objednávky
- **Transaction** (Transakce)
 - Zjednodušení, viz Milník
 - *http://[domain]/[type]/[id]/[versionId]/transaction/[transactionId]*
 - * Transakce u dokumentu
 - *http://[domain]/order/[orderId]/transaction/[transactionId]*
 - * Transakce u objednávky
- **Party** (Smluvní strana) - *http://[domain]/party/[partyId]*
- **Address** (Adresa) - *http://[domain]/party/[partyId]/address*
 - Jedná se o podřízenou položku smluvní strany
- **SuperiorInstitution** (Nadřazená instituce) - *http://[domain]/superiorInstitution/[superiorInstitutionId]*

4.2 Ontologie pro publikaci dat o smlouvách

Než začneme s tvorbou ontologie, je dobré si uvědomit, že vycházíme z již hotového datového standardu. Nemáme tedy v tvorbě úplnou svobodu. Cílem tedy bude tvorba takové ontologie, která bude odpovídat stávajícímu datovému standardu, a přesto se bude snažit využít co nejvíce již existujících ontologií.

Samotnou tvorbu ontologie rozdělíme do dvou kroků:

- Analýza vhodných, již existujících ontologií
- Tvorba samotné ontologie

4.2.1 Analýza vhodných, již existujících ontologií

Při tvorbě ontologie se zaměříme na otázku, zdali existuje třída, či predikát v nějaké ontologii sémanticky ekvivalentní třídě, či konkrétní položce datového standardu pro smlouvy. Takových vhodných ontologií ve světě Linked Data může být celá řada. K výběru stačí libovolná z nich.

V následujícím seznamu je výčet vybraných ontologií¹, které se jeví jako vhodné pro použití při popisování smluv². U každého bodu je zmíněn popis ontologie, důvod, proč byla daná ontologie zvolena, a seznam tříd a predikátů vhodných k použití.

Vybrané ontologie:

- **Commerce (com)** (<https://w3id.org/commerce#>)[52] - ontologie pro popisování obchodních transakcí
 - Důvod použití - užitečná třída transakce
 - Vybrané třídy
 - * Transaction - třída reprezentující transakci
 - Vybrané predikáty
 - * source - zdroj transakce
 - * destination - cíl transakce
- **DublinCore (dc,dcmi)** (<http://purl.org/dc/terms/>)[26] - základní ontologie pro popis metadat
 - Důvod použití - základní a všeobecně známá ontologie popisující metadata
 - Vybrané predikáty
 - * created - datum vytvoření
 - * creator - tvůrce

¹V závorce u názvu ontologií je uvedena typicky používaná zkratka

²Obecně výběr ontologií nemusíme považovat za striktní. Každou třídu, či predikát lze označit jako sémanticky ekvivalentní jiné třídě, či predikátu. Slouží k tomu konstrukce jazyka Owl - *equivalentClass*, resp. *equivalentProperty*.

- * date - obecné datum
 - * description - popis metadat
 - * identifier - jednoznačný identifikátor
 - * issued - datum publikace
 - * language - jazyk
 - * modified - datum modifikace
 - * publisher - vydavatel
 - * rights - licence
 - * title - název dokumentu
 - * type - typ dokumentu
- **Friend-of-a-Friend (foaf)** (<http://xmlns.com/foaf/0.1>)[27] - ontologie pro popis vazeb mezi lidmi
 - Důvod použití - vhodná pro označení třídy vydavatele
 - Vybrané třídy
 - * Person - třída reprezentující osobu
 - * Organization - třída reprezentující organizaci
 - Vybrané predikáty
 - * name - jméno osoby
 - * mbox - email osoby
- **GoodRelations (gr)** (<http://purl.org/goodrelations/v1#>)[53] - ontologie pro popis produktů, cen a obchodních dat
 - Důvod použití - známá ontologie, vhodná pro popis smluvních stran a informací o cenách
 - Vybrané třídy
 - * BusinessEntity - třída popisující hospodářské subjekty
 - * PriceSpecification - třída popisující informace o ceně
 - Vybrané predikáty
 - * hasCurrency - měna
 - * hasCurrencyValue - cena
 - * legalName - název subjektu
 - * valueAddedTaxIncluded - plátce DPH
- **Schema** (<http://schema.org/>)[28] - obecná ontologie mající za cíl pokrývat co největší možné množství informací
 - Důvod použití - známá ontologie, možnost využití pro popis adresních údajů
 - Vybrané třídy
 - * PostalAddress - třída reprezentující adresu
 - Vybrané predikáty

- * url - URL adresa obsahu
 - * address - adresa
 - * addressCountry - země
 - * addressLocality - město
 - * postalCode - PSČ
 - * streetAddress - ulice
- **Vann** (<http://purl.org/vocab/vann/>)[54] - anotační ontologie pro dokumenty
 - Důvod použití - nespojuje s datovým standardem, tato ontologie je vhodná pro popis ontologií a bude zmíněna níže v publikaci [3].
 - Vybrané predikáty
 - * preferredNamespaceUri - preferovaná adresa ontologie
 - * preferredNamespacePrefix - preferovaná zkratka
 - * usageNote - poznámka k použití

4.2.2 Tvorba ontologie

Každou položku datového standardu namapujeme na třídu, či predikát výsledné ontologie³. Pro některé položky využijeme zmíněné predikáty z již existujících ontologií, pro ostatní vytvoříme třídy a predikáty vlastní.

Vlastní ontologii nazveme jako open-contracting a budeme pro ni používat zkratku cn.

Výsledné mapování můžeme vidět v následujících tabulkách 4.1 - 4.14. V prvním sloupečku se nachází entita/vlastnost datového standardu, v druhém namapovaná třída/predikát a v třetím případná poznámka⁴.

Dokument

Entita/Vlastnost	Třída/Predikát	Poznámka
Document	cn:Document	
uri	cn:uri	
document	schema:url	
versions	cn:version	
type	dcmi:type	
publisher	dc:publisher	Odkaz na objekt vydavatele
valid	cn:valid	
plainText	cn:plainText	
responsiblePersons	cn:responsiblePerson	Původně kolekce
anonymised	cn:anonymised	

Tabulka 4.1: Mapování entity Document

Vydavatel

Entita/Vlastnost	Třída/Predikát	Poznámka
Publisher	foaf:Organization	
id	dc:identifier	
name	gr:legalName	
noID	cn:noID	
country	schema:addressCountry	
authentication	cn:authentication	

Tabulka 4.2: Mapování entity Vydavatel

³Kolekce jsou řešeny výčtem. Např. kolekce odpovědných osob bude vypadat jako výčet trojic se stejným odpovídajícím predikátem reprezentující odpovědnou osobu.

⁴Entity jsou uváděny velkým písmem, vlastnosti malým

Verze

Entita/Vlastnost	Třída/Predikát	Poznámka
Version	cn:Version	
publisherId	cn:publisherId	
version	cn:versionOrder	
uri	cn:uri	
published	dc:issued	

Tabulka 4.3: Mapování entity Verze smlouvy

Smlouva

Entita/Vlastnost	Třída/Predikát	Poznámka
Contract	cn:Contract	
awardID	cn:awardID	
awardProfileID	cn:awardProfileID	
amount	cn:amount	Pro lepší zapouzdření informací o ceně definujeme cn:amount jako objekt typu gr:PriceSpecification obsahující informace o ceně amountValue (gr:hasCurrencyValue) a currency (gr:hasCurrency). Nedoporučuje se tedy pak používat položku currency zvlášť.
amountNoVat	cn:amountNoVat	Viz amount
title	dc:title	
contractType	cn:contractType	
parties	cn:party	Původně kolekce
subjectType	cn:subjectType	
priceAnnual	cn:priceAnnual	
currency	gr:hasCurrency	
dateSigned	dc:created	
validFrom	cn:validFrom	
validUntil	cn:validUntil	
funding	cn:funding	
attachments	cn:attachment	Původně kolekce, odkazy na přílohy
amendments	cn:amendment	Původně kolekce, odkazy na dodatky
competency	cn:competency	
currentValidContract	cn:currentValidContract	
description	dc:description	
implementation	cn:implementation	Odkaz na objekt implementace

Tabulka 4.4: Mapování entity Smlouva

Příloha

Entita/Vlastnost	Třída/Predikát	Poznámka
Attachment	cn:Attachment	
title	dc:title	
contract	cn:contract	
attachmentOrder	cn:attachmentOrder	

Tabulka 4.5: Mapování entity Příloha

Dodatek

Entita/Vlastnost	Třída/Predikát	Poznámka
Amendment	cn:Amendment	
title	dc:title	
contract	cn:contract	
amendmentOrder	cn:amendmentOrder	
dateSigned	dc:created	

Tabulka 4.6: Mapování entity Dodatek

Smluvní strana

Entita/Vlastnost	Třída/Predikát	Poznámka
Party	gr:BusinessEntity	
id	dc:identifier	
localID	cn:localID	
name	gr:legalName	
payer	cn:payer	
noID	cn:noID	
country	schema:addressCountry	
address	schema:address	Odkaz na objekt adresy
paysVAT	cn:paysVAT	
superiorInstitution	cn:superiorInstitution	

Tabulka 4.7: Mapování entity Smluvní strana

Nadřazená instituce

Entita/Vlastnost	Třída/Predikát	Poznámka
SuperiorInstitution	gr:BusinessEntity	
id	dc:identifier	
localID	cn:localID	
name	gr:legalName	
noID	cn:noID	
country	schema:addressCountry	

Tabulka 4.8: Mapování entity Nadřazená instituce

Adresa

Entita/Vlastnost	Třída/Predikát	Poznámka
Address	schema:PostalAddress	
streetAddress	schema:streetAddress	
locality	schema:addressLocality	
postalCode	schema:postalCode	
nuts	cn:nuts	

Tabulka 4.9: Mapování entity Nadřazená instituce

Objednávka

Entita/Vlastnost	Třída/Predikát	Poznámka
Order	cn:Order	
parentDocument	cn:parentDocument	Odkaz na rodičovský dokument
subjectType	cn:subjectType	
parties	cn:party	Původně kolekce
title	dc:title	
amount	cn:amount	Viz amount u smlouvy
currency	gr:hasCurrency	
dateSigned	dc:created	
implementation	cn:implementation	

Tabulka 4.10: Mapování entity Objednávka

Faktura

Entita/Vlastnost	Třída/Predikát	Poznámka
Invoice	cn:Invoice	
parentDocument	cn:parentDocument	Odkaz na rodičovský dokument
parties	cn:party	Původně kolekce
title	dc:title	
amount	cn:amount	Viz amount u smlouvy
currency	gr:hasCurrency	
dateSigned	dc:created	
dueDate	cn:dueDate	

Tabulka 4.11: Mapování entity Faktura

Rozšiřující entity

Implementace

Entita/Vlastnost	Třída/Predikát	Poznámka
Implementation	cn:Implementation	
milestones	cn:milestone	Původně kolekce, odkazy na milníky
transactions	cn:transaction	Původně kolekce, odkazy na transakce

Tabulka 4.12: Mapování entity Implementace

Milník

Entita/Vlastnost	Třída/Predikát	Poznámka
Milestone	cn:Milestone	
title	dc:title	
dueDate	cn:dueDate	

Tabulka 4.13: Mapování entity Milník

Transakce

Entita/Vlastnost	Třída/Predikát	Poznámka
Transaction	com:Transaction	
date	dc:date	
amount	cn:amount	Viz amount u smlouvy
senderOrganization	com:source	
receiverOrganization	com:destination	
publisherId	cn:publisherId	

Tabulka 4.14: Mapování entity Transakce

4.2.3 Publikace

K serializaci výsledné ontologie využijeme formátu Turtle. Soubor se skládá z hlavičky a definicí nově vytvořených tříd a predikátů. V hlavičce definujeme prefixy použitých ontologií a základní informace o ontologii. Použité třídy a predikáty zmíníme v poznámkách k použití (predikát *vann:usageNote*). Celou ontologie lze nalézt v Příloze D.

4.3 Možnosti propojení na související data

První bezpochyby zajímavou možností je propojení smluv s veřejnými zakázkami, resp. věstníkem veřejných zakázek provozovaným Ministerstvem pro místní rozvoj. Jednotlivé smlouvy mající spojitost s veřejnou zakázkou poznáme podle vlastnosti AwardID, resp. evidenční číslo veřejné zakázky.

Dalšími zajímavými prvky k propojení jsou pokročilé informace o smluvních stranách. Každá smluvní strana vystupující ve smlouvě může mít zveřejněno mimo jiné identifikační číslo a adresu. Nabízí se tedy propojení s národními registry ARES a RÚIAN. ARES je registrem informací o ekonomických subjektech provozovaný Ministerstvem financí, RÚIAN je registrem územní identifikace, adres a nemovitostí provozovaný Českým úřadem zeměměřickým a katastrálním. Vhodné informace k propojení tedy jsou:

- Evidenční číslo veřejné zakázky u smlouvy s věstníkem veřejných zakázek
 - Iniciativa OpenData.cz[8] zpracovává údaje o veřejných zakázkách v RDF podobě, využijeme tedy propojení právě s tímto datasetem
 - Cílové URL -
`http://linked.opendata.cz/resource/domain/buyer-profiles/contract/-cz/{EvidencniCisloZakazky}`
- Identifikační číslo smluvní strany s možností propojení na ARES
 - OpenData.cz aktuálně zpracovávají také údaje z registru ARES
 - Cílové URL -
`http://linked.opendata.cz/resource/business-entity/CZ{ICO}`⁵
- Adresa smluvní strany na RÚIAN
 - V rámci iniciativy OpenData.cz vzniká služba na mapování adres na RÚIAN. V budoucnu proto bude možné využít i této možnosti propojení.

4.4 Provázání s datovým formátem JSON

V třetí kapitole jsme si ukázali, jak publikovat smlouvy v datovém formátu JSON. Vzhledem k budoucímu možnému využití v aplikacích pracujících nad smlouvami v JSON dokumentech by bylo dobré nastínit, jak taková data rozšířit, aby se z nich stala zároveň RDF data. Cílem je ale zachovat původní strukturu JSON souboru, resp. aby data byla validní vůči JSON schématu. Pro tyto účely je ideální formát JSON-LD. Jediné, co nám stačí, je v původních datech každému objektu přiřadit *@id*, *@type* a definovat *@context* (viz první kapitola Příklad serializovaných dat ve formátu JSON-LD). Při tvorbě ontologie jsme si popsali mapování entit a položek z datového standardu na konkrétní třídy a predikáty. V kontextu je tedy přesně takovéto mapování, viz příklad kódu 4.1. Na příkladu kódu 4.2 je již vidět výsledný JSON-LD soubor s jednou smlouvou a dvěma

⁵Informace skrývající se za tímto odkazem jsou sjednocením více datových zdrojů, ne pouze ARESu

smluvními stranami (pro porovnání, původní JSON soubor viz kód 3.1). Jedná se tedy o RDF data, která popisuje námi definovaná ontologie a zároveň jde o data splňující datový standard, resp. jsou validní vůči JSON Schématu. Hlavním přínosem je to, že RDF data serializovaná v takto definovaném JSON-LD formátu budou použitelná v budoucích aplikacích, příp. registru pracujícím nad datovým standardem.

```

1 {
2   "@context": {
3
4     "cn": "http://tiny.cc/open-contracting#",
5     "com": "https://w3id.org/commerce#",
6     "dc": "http://purl.org/dc/terms/",
7     "dcmi": "http://dublincore.org/documents/dcmi-type-vocabulary/",
8     "foaf": "http://xmlns.com/foaf/0.1/",
9     "gr": "http://purl.org/goodrelations/v1#",
10    "owl": "http://www.w3.org/2002/07/owl#",
11    "schema": "http://schema.org/",
12    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
13    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
14    "xsd": "http://www.w3.org/2001/XMLSchema#",
15
16    "id" : "dc:identifier",
17    "language" : "dc:language",
18
19    "Contract" : "cn:Contract",
20    "Attachment" : "cn:Attachment",
21    "Amendment" : "cn:Amendment",
22    "Order" : "cn:Order",
23    "Invoice" : "cn:Invoice",
24    "Publisher" : "foaf:Organization",
25    "Version" : "cn:Version",
26    "Party" : "gr:BusinessEntity",
27    "SuperiorInstitution" : "gr:BusinessEntity",
28    "Address" : "schema:PostalAddress",
29    "Implementation" : "cn:Implementation",
30    "Milestone" : "cn:Milestone",
31    "Transaction" : "com:Transaction",
32    "PriceSpecification" : "gr:PriceSpecification",
33
34    "documents" : { "@id": "cn:documents", "@container": "@set" },
35    "orders" : { "@id": "cn:orders", "@container": "@set" },
36    "invoices" : { "@id": "cn:invoices", "@container": "@set" },
37
38    "uri" : { "@id": "cn:uri", "@type" : "@id" },
39    "document" : { "@id": "schema:url", "@type" : "@id" },
40    "valid" : { "@id": "cn:valid", "@type" : "xsd:boolean" },
41    "plainText" : "cn:plainText",
42    "anonymised" : { "@id": "cn:anonymised", "@type" : "xsd:boolean"
43    },
44    "responsiblePersons" : { "@id": "cn:responsiblePerson", "@container": "@set" },
45    "attachments" : { "@id": "cn:attachment", "@container": "@set"
46    },
47    "amendments" : { "@id": "cn:amendment", "@container": "@set" },
48
49    "amount": "cn:amount",
50    "amountNoVat": "cn:amountNoVat",

```



```

49   "amountValue": { "@id": "gr:hasCurrencyValue", "@type": "
xsd:float" },
50   "currency": "gr:hasCurrency",
51
52   "awardID" : "cn:awardID",
53   "awardProfileID" : "cn:awardProfileID",
54   "title" : "dc:title",
55   "type" : { "@id": "demi:type", "@container": "@set" },
56   "contractType" : "cn:contractType",
57   "subjectType" : "cn:subjectType",
58   "priceAnnual" : { "@id": "cn:priceAnnual", "@type": "
xsd:boolean" },
59   "dateSigned" : { "@id": "dc:created", "@type": "xsd:date" },
60   "validFrom" : { "@id": "cn:validFrom", "@type": "xsd:date" },
61   "validUntil" : { "@id": "cn:validUntil", "@type": "xsd:date" },
62   "funding" : "cn:funding",
63   "currentValidContract" : { "@id": "cn:currentValidContract", "
@type": "@id" },
64   "description" : "dc:description",
65   "competency" : { "@id": "cn:competency", "@container": "@set" },
66   "parties" : { "@id": "cn:party", "@container": "@set" },
67
68   "localID" : { "@id": "cn:localID", "@type": "xsd:integer" },
69   "name" : "gr:legalName",
70   "payer" : { "@id": "cn:payer", "@type": "xsd:boolean" },
71   "noID" : { "@id": "cn:noID", "@type": "xsd:boolean" },
72   "country" : "schema:addressCountry",
73   "paysVAT" : { "@id": "cn:paysVAT", "@type": "xsd:boolean" },
74   "superiorInstitution" : "cn:superiorInstitution",
75
76   "publisher" : "dc:publisher",
77   "authentication" : "cn:authentication",
78
79   "address" : "schema:address",
80   "streetAddress" : "schema:streetAddress",
81   "locality" : "schema:addressLocality",
82   "postalCode" : { "@id": "schema:postalCode", "@type": "
xsd:integer" },
83   "nuts" : "cn:nuts",
84
85   "versions" : { "@id": "cn:version", "@container": "@set" },
86   "version" : { "@id": "cn:versionOrder", "@type": "xsd:integer"
},
87   "published" : { "@id": "dc:issued", "@type": "xsd:dateTime" },
88
89   "implementation" : "cn:implementation",
90   "milestones" : { "@id": "cn:milestone", "@type": "@id", "
@container": "@set" },
91   "transactions" : { "@id": "cn:transaction", "@type": "@id", "
@container": "@set" },
92   "dueDate" : { "@id": "cn:dueDate", "@type": "xsd:dateTime" },
93   "date" : { "@id": "dc:date", "@type": "xsd:dateTime" },
94   "publisherId" : "cn:publisherId",
95   "senderOrganization" : { "@id": "com:source", "@type": "@id" },
96   "receiverOrganization" : { "@id": "com:destination", "@type": "
@id" },
97
98   "contract" : { "@id": "cn:contract", "@type": "@id" },

```

```

99   "parentDocument" : { "@id": "cn:parentDocument", "@type": "
    @id" },
100
101   "attachmentOrder" : { "@id": "cn:attachmentOrder", "@type": "
    xsd:integer" },
102   "amendmentOrder" : { "@id": "cn:amendmentOrder", "@type": "
    xsd:integer" }
103 }
104 }

```

Výpis kódu 4.1: JSON-LD Context

```

1 {
2   "@context": "http://tiny.cc/open-contracting-context",
3
4   "@id" : "http://rsmluv.cz/data/89f689cd-e784-4374-bb17-94144679
    d46f",
5   "id" : "89f689cd-e784-4374-bb17-94144679d46f",
6   "published": "2014-03-25T23:20:50+01:00",
7   "language": "cs",
8
9   "documents": [
10    {
11      "@id" : "http://rsmluv.cz/contract/12345/2",
12      "@type" : "Contract",
13      "uri" : "http://rsmluv.cz/contract/12345/2",
14      "document" : "http://rsmluv.cz/file/b15a3c45-5595-4a28-b156
        -4578edeb2a98/Smlouva12345.docx",
15      "type" : "Smlouva",
16      "valid" : true,
17      "anonymised" : false,
18
19      "awardID": "486026",
20      "awardProfileID": "OI-010143",
21      "title": "Brno, Vackova, Šafaříkova – rekonstrukce kanalizace
        a vodovodu",
22      "contractType": "Kupní smlouva",
23      "subjectType": "Právní, finanční překladatelské, pojišťovnické
        , poradenské a jiné služby",
24      "priceAnnual": false,
25      "dateSigned": "2011-11-16",
26      "validFrom": "2011-11-02",
27      "validUntil": "2012-06-30",
28      "competency" : [ "Soukromoprávní smlouva" ],
29      "currentValidContract": "http://zakazky.brno.cz/?pg=detail&id
        =18249&list=135",
30      "description": "Projektová dokumentace pro stavební povolení a
        zadání stavby bude řešit rekonstrukci stávající kanalizační
        stoky z profilu DN 500 na DN 800/1200 v délce 146 m, rekonstrukci
        kanalizačních přípojek pod veřejným prostranstvím a přepojení vš
        ech stávajících dešťových vpustí a také vybourání vozovek a chodn
        íků nad rýhou a zásyp rýhy recyklátem. Součástí bude inženýrsko-
        geologický průzkum, geodetické zaměření dotčené oblasti,
        inventarizace zeleně, výkaz výměr, položkový rozpočet a výkon
        autorského dozoru až do dokončení stavby. Dokumentace bude
        projednána s orgány státní správy a s účastníky stavebního řízení
        a jejich připomínky budou do dokumentace zapracovány.",
31

```

```

32     "responsiblePersons" : [ "Ing. Petr Vokřál ", "Mgr. Adriana
    Krnáčová, MBA" ],
33
34     "amount" : {
35         "amountValue" : 584520.00,
36         "currency" : "CZK"
37     },
38
39     "publisher" : {
40         "@id" : "http://rsmluv.cz/contract/12345/2/publisher" ,
41         "@type" : "Publisher" ,
42         "id" : "6003508" ,
43         "name" : "Statutární město Brno" ,
44         "noID" : false ,
45         "country" : "CZE" ,
46         "authentication" : "email"
47     },
48
49     "parties" : [
50         { "@id" : "http://rsmluv.cz/party/132456" },
51         { "@id" : "http://rsmluv.cz/party/987654" }
52     ],
53
54     "implementation" : {
55         "@id" : "http://rsmluv.cz/contract/12345/2/implementation" ,
56         "@type" : "Implementation" ,
57
58         "milestones" : [
59             {
60                 "@id" : "http://rsmluv.cz/contract/132456/2/milestone
    /5830" ,
61                 "@type" : "Milestone" ,
62                 "title" : "Výpověď smlouvy" ,
63                 "dueDate" : "2012-06-20T23:20:50+01:00"
64             }
65         ],
66         "transactions" : [
67             {
68                 "@id" : "http://rsmluv.cz/contract/132456/2/transaction
    /132456" ,
69                 "@type" : "Transaction" ,
70                 "publisherId" : "1269483" ,
71                 "date" : "2012-01-01T18:35:20+01:00" ,
72                 "amount" : 300000,
73                 "senderOrganization" : "http://rsmluv.cz/party/987654" ,
74                 "receiverOrganization" : "http://rsmluv.cz/party/987654"
75             },
76             {
77                 "@id" : "http://rsmluv.cz/contract/132456/2/transaction
    /934584" ,
78                 "@type" : "Transaction" ,
79                 "publisherId" : "934584" ,
80                 "date" : "2012-02-01T09:13:40+01:00" ,
81                 "amount" : 284520,
82                 "senderOrganization" : "http://rsmluv.cz/party/987654" ,
83                 "receiverOrganization" : "http://rsmluv.cz/party/987654"
84             }
85         ]

```

```

86     },
87
88     "versions" : [
89     {
90         "version" : 1,
91         "@id" : "http://rsmluv.cz/contract/12345/1/version",
92         "@type" : "Version",
93         "uri" : "http://rsmluv.cz/contract/12345/1",
94         "published" : "2014-09-15T23:20:50+01:00"
95     },
96     {
97         "version" : 2,
98         "@id" : "http://rsmluv.cz/contract/12345/2/version",
99         "@type" : "Version",
100        "uri" : "http://rsmluv.cz/contract/12345/2",
101        "published" : "2015-03-15T14:35:28+01:00"
102    }
103 ]
104 }
105 ],
106
107 "parties" : [
108 {
109     "@id" : "http://rsmluv.cz/party/132456",
110     "@type" : "Party",
111     "localID" : 132456,
112     "id" : "44992785",
113     "name" : "Statutární město Brno",
114     "payer" : false,
115     "noID" : false,
116     "country" : "CZE",
117     "address" : {
118         "@id" : "http://rsmluv.cz/party/132456/address",
119         "@type" : "Address",
120         "streetAddress" : "Dominikánské náměstí 196/1",
121         "locality" : "Brno-město, Brno",
122         "postalCode" : 60200,
123         "nuts" : "CZ064"
124     },
125     "superiorInstitution" : {
126         "@id" : "http://rsmluv.cz/superiorInstitution/00064581",
127         "@type" : "SuperiorInstitution",
128         "id" : "00064581",
129         "localID" : 56486,
130         "name" : "Magistrát hlavního města Prahy",
131         "noID" : false,
132         "country" : "CZE"
133     }
134 },
135 {
136     "@id" : "http://rsmluv.cz/party/987654",
137     "@type" : "Party",
138     "localID" : 987654,
139     "id" : "46347011",
140     "name" : "Kovoprojekta Brno a.s.",
141     "payer" : true,
142     "noID" : false,
143     "country" : "CZE",

```

```
144     "address" : {
145         "@id" : "http://rsmluv.cz/party/987654/address",
146         "@type" : "Address",
147         "streetAddress" : "Šumavská 416/15",
148         "locality" : "Ponava, Brno",
149         "postalCode" : 60200,
150         "nuts" : "CZ064"
151     }
152 }
153 ]
154
155 }
```

Výpis kódu 4.2: JSON-LD Soubor s jednou smlouvou

5. Požadavky na platformu pro otevřené smlouvy

Jedním z cílů této práce je návrh a implementace platformy pro otevírání smluv veřejných institucí. Platforma bude mít tři základní části. První částí je převod interně uložených smluv v relačních databázích do otevřeného formátu splňujícího principy Linked Data. Druhou je jednotné úložiště smluv v otevřeném formátu a třetí je nad tímto úložištěm zpřístupnit otevřené smlouvy koncovým uživatelům. Shrňme si v následující části základní funkční a nefunkční požadavky na platformu:

5.1 Funkční požadavky

- Platforma bude umět převádět údaje o smlouvách z relačních databází veřejných institucí do otevřených dat
 - Jedná se o otevřená data, tedy dostupná on-line ke stažení/prohlížení
 - Otevřená data by měla splňovat datový standard pro otevřené smlouvy (kapitola Otevřené smlouvy)
 - Otevřená data by měla splňovat principy Linked Data (kapitola Otevřená data a principy Linked Data), (Otevřené smlouvy jako Linked Data))
 - Výstupní formát by měl reflektovat alespoň jeden publikační formát datového standardu pro budoucí kompatibilitu
- Platforma bude umět ukládat otevřené smlouvy z mnoha zdrojů v jednotném úložišti
 - Jednotné úložiště bude reflektovat, že se jedná o Linked Data, resp. bude mít formu databáze trojic (Triplestore)
 - Data budou dostupná skrze API
- Platforma nad jednotným úložištěm zpřístupní otevřené smlouvy koncovým uživatelům formou webové aplikace
 - Aplikace by měla umět zobrazovat seznam smluv
 - Aplikace by měla umět filtrovat smlouvy podle jednotlivých veřejných institucí
 - Aplikace by měla umožňovat prohlédnout si detail veřejné instituce a její smlouvy
 - Aplikace by měla umožňovat zobrazit detail smlouvy

5.2 Nefunkční požadavky

- Použitelnost
 - Platforma by měla počítat s použitím v českém prostředí
 - Platforma bude obsahovat uživatelskou dokumentaci
 - Webová aplikace by měla vhodným způsobem demonstrovat výhody Linked Open Data
- Výkon
 - Konverzní mechanismus platformy by měl umět zpracovávat i větší objemy dat (řádově alespoň desetitisíce údajů v relačních databázích)
- Rozšiřitelnost a modifikovatelnost
 - Každá ze tří částí platformy by měla být snadno nahraditelná a modifikovatelná
- Integrovatelnost
 - Nasazení konverzního mechanismu u veřejných institucí by mělo být co nejjednodušší
- Bezpečnost
 - Při použití konverzního mechanismu by nemělo dojít k narušení vnitřních dat veřejných institucí

6. Návrh platformy pro otevřené smlouvy

6.1 Architektura

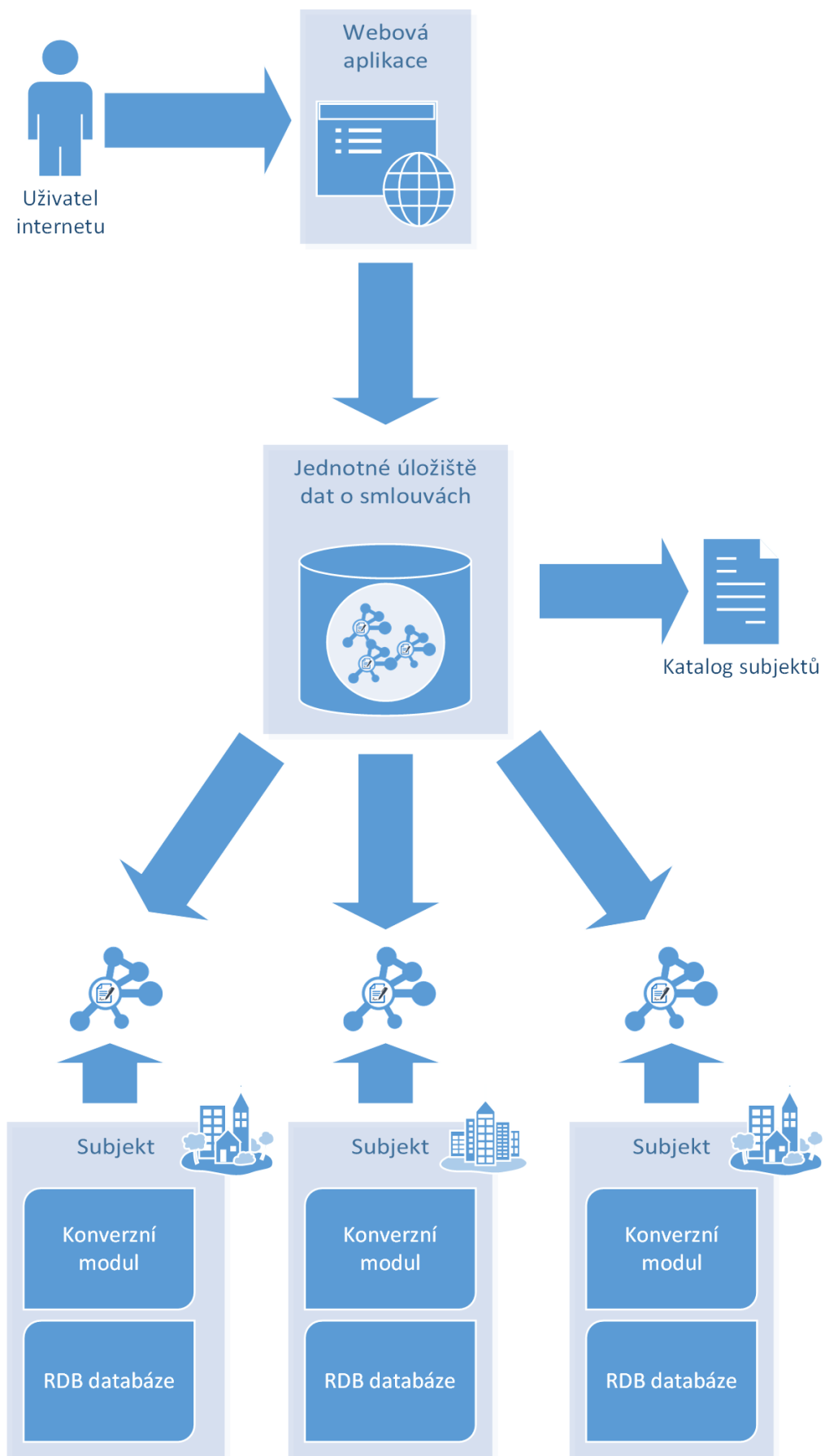
Základem pro platformu pro otevřené smlouvy jsou tři nezávislé moduly. Konverzní modul, jednotné úložiště a webová aplikace. Každá zapojená veřejná instituce bude mít svůj konverzní modul. Jednotlivé konverzní moduly budou pracovat nad konkrétními relačními databázemi jednotlivých veřejných institucí a publikovat smlouvy jako Linked Open Data. Tato data pak budou ukládána v jednotném úložišti.

Základní otázkou, kterou si je třeba položit, je způsob shromažďování dat. První možností je směr centralizovaný, resp. každý konverzní modul publikovaná data pošle do jednotného úložiště. Druhým přístupem je decentralizace, kdy konverzní moduly vystaví data na internetu a registrují přístup k datům v datovém katalogu. Jednotné úložiště si poté taková data na základě katalogu obstarává samo. Výhoda první možnosti spočívá v snadnější udržitelnosti správy nad jednotlivými instancemi veřejných institucí. Subjekty mohou v určitém intervalu posílat převedené smlouvy do úložiště bez dalších nároků na správu. V druhém přístupu jsou naopak zvýšeny nároky na jednotlivé subjekty, které vypublikované smlouvy samy musejí udržovat. Výhodou ale naopak je, že každý subjekt se stane lokálním úložištěm smluv, na které se lze odkazovat, např. na webových stránkách subjektu, vytvářet nad daty aplikace apod. První varianta je vhodná pro myšlenku centralizovaného registru, který pouze definuje podmínky, jak nahrávané smlouvy mají vypadat. Subjekty potom mají svobodu v tom, jak data převedou, pošlou, případně vypublikují. Druhá možnost naopak podporuje myšlenku, že každý subjekt zveřejňuje pouze své smlouvy a po vypublikování je umožní komukoli využít. Není proto překvapením, že myšlenky otevřených a propojených dat vyhovuje více druhý přístup.

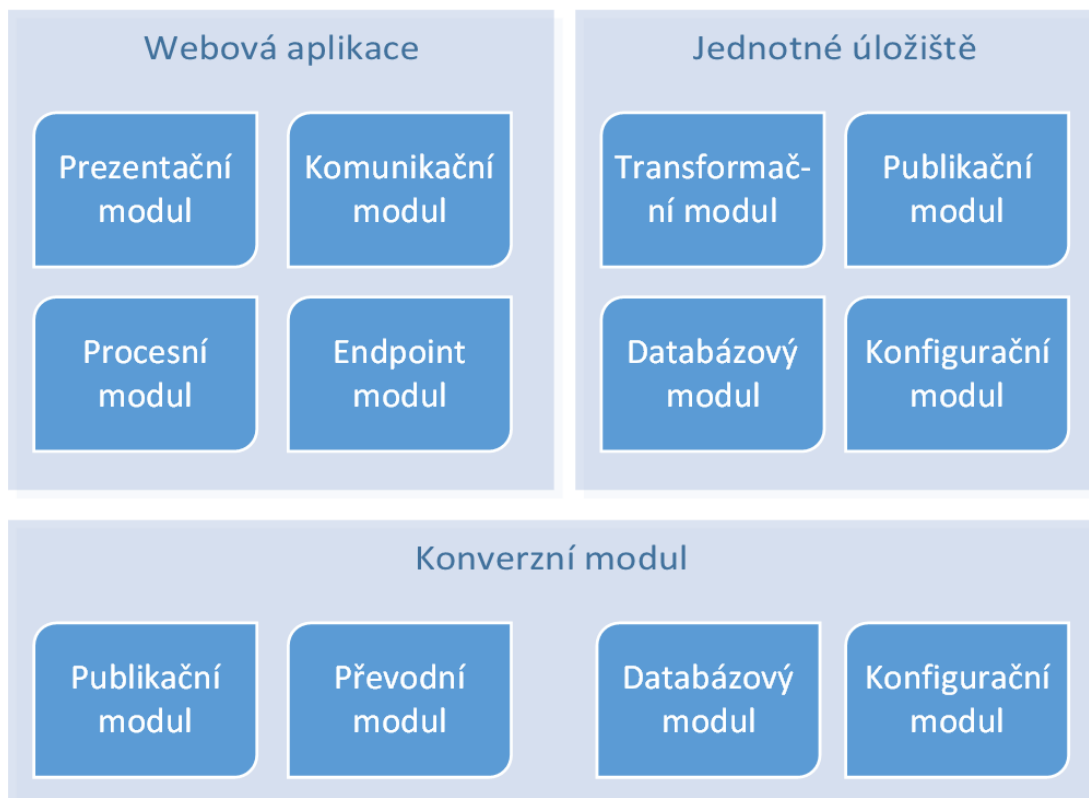
K efektivnímu naplnění druhého přístupu je důležité, že součástí platformy bude konverzní modul, který převod za subjekty řeší a celý proces otevírání smluv pomůže zautomatizovat. Částečně tak vyřešíme nevýhodu druhého přístupu a část agendy otevírání smluv přebere platforma. Na konkrétním subjektu tak zbývá pouze udržovat v chodu databázi a server (typicky webový), kde konverzní modul poběží.

Situaci můžeme dále zjednodušit tím, že subjekt zpřístupní svou databázi online. Konverzní modul tak bude moci pracovat nezávisle na subjektu a data pouze vzdáleně načítat. Narážíme zde ale na bezpečnostní limity. Databáze veřejných institucí bude typicky přístupná pouze v rámci privátní sítě. Řešením by mohlo být, že server, kde poběží konverzní modul, bude připojen do privátní sítě subjektu, druhou možností je např. klonování databáze smluv do veřejně přístupné databáze. Konkrétní řešení přístupu k databázi však necháme na konkrétních potřebách jednotlivých subjektů. Řekněme tedy, že konverzní modul platformy bude vyžadovat pouze připojení ke konkrétní databázi.

Základní pohled na platformu můžeme vidět na Obr. 6.1.



Obrázek 6.1: Základní pohled na platformu otevřených smluv (Logical view)



Obrázek 6.2: Rozdělení platformy do modulů (Decomposition view)

6.2 Konverzní mechanismus

Návrh konverzního mechanismu rozdělíme do čtyř modulů (viz Obr. 6.2). V rámci databázového modulu bude docházet ke komunikaci mezi připojenou databází a zbytkem konverzního mechanismu. V konfiguračním modulu se očekává definování základních vlastností konverzního mechanismu, převážně potřebné vstupní údaje pro převodní modul.

Převodní modul

Účelem tohoto modulu je převod relačních dat do RDF podoby. Nejjednodušším přístupem je manuální konverze, resp. ruční tvorba RDF výstupů. Tento přístup může být vhodný pro úzce specifické situace, avšak pro obecný přístup a možnost znovupoužití konverzního modulu vhodný není. Další možností je tvorba paralelní triplestore databáze. Relační data bychom v určitých intervalech převáděli z jedné do druhé. Tento přístup je výhodný, pokud chceme dosáhnout robustního řešení, budovat vlastní RDF úložiště, obohacovat data, přidávat další datasety apod. Klade však velké softwarové i hardwarové nároky na subjekt, vyžaduje netriviální údržbu a jedná se v podstatě o duplikovaná data z relační databáze. Poslední možností, kterou uvedeme, je tvorba wrapperu nad relační databází. Máme-li požadavek na RDF data, wrapper ho převede na ekvivalentní SQL dotaz do databáze a vrácená data převede zpět do odpovídající RDF podoby. Hlavní výhodou je, že takto uložená data jsou pouze v relační databázi a RDF data jsou tak vždy aktuální. Cílem je řešení s co nejmenší zátěží pro subjekt a s co

nejlepším usnadněním znovupoužitelnosti. Ideálním řešením se tedy jeví tvorba wrapperu.

Požadovanou funkcionalitou wrapperu je namapování datového modelu relační databáze na datový standard pro otevřené smlouvy. Mezi jazyky sloužící k popsání konkrétního mapování patří např. jazyk R2RML[55], nebo D2RQ[56]. V rámci MFF UK vznikají projekty pracující nad R2RML (s potenciálem využití), jazyk je navíc doporučeným standardem konsorcia W3C. Zvolíme tedy jazyk R2RML¹.

Publikační modul

Převedená data můžeme publikovat třemi základními způsoby:

- Dump - veškerá data jsou zpřístupněna formou stažitelného souboru serializovaného v nějakém z RDF formátů
- Dereferencovaná URI jednotlivých entit - každá entita je dostupná pod svým URI, typicky ve formě HTML stránky
- API - v kontextu RDF se typicky jedná o webovou službu ve formě SPARQL endpointu umožňující libovolné dotazování nad daty.

Naší snahou je, aby vypublikovaná data mohly využívat i jiné aplikace, než pouze jednotné úložiště v rámci platformy. K tomu je ideální API. Zvolíme proto SPARQL endpoint.

Pro naplnění principů Linked Data ale potřebujeme vyřešit dereferencování URI entit. Nad SPARQL endpointem se dereference provede jednoduše tak, že každé HTTP URI odkazující na konkrétní entitu se převede na vhodný SPARQL dotaz vracející požadovaná data.

Data budou publikována jak ve formě HTML stránky, tak v RDF formátech. Nutným základem bývá formát N-Triples a Turtle. Určíme podmínku, že pro dump je nutné umožnit i serializaci ve formátu JSON-LD z důvodu budoucí kompatibility s datovým standardem.

Anonymizace

Typickým problémem s publikací dat je, že mohou podléhat zákonu o ochraně osobních údajů[57]. Některá data je proto před zveřejněním nutné anonymizovat. Proces a řešení anonymizace není předmětem této práce. Řekněme, že platforma počítá s tím, že subjekt si anonymizaci údajů vyřeší na své straně.

6.3 Jednotné úložiště

Agendou jednotného úložiště bude sbírat data vypublikovaná jednotlivými subjekty. Zapojené subjekty budeme řešit formou datového katalogu, v kterém se

¹Principem jazyka R2RML je mapování tabulek pomocí tzv. „triples map“. Jedná se o pravidlo v rámci kterého se definuje nad konkrétní tabulkou mapování subjektu a následně sada pravidel pro mapování dvojic predikát/objekt. Typicky pak subjekt reprezentuje konkrétní tabulku, predikáty jednotlivé sloupce a objekty uložené hodnoty.[55]

budou nacházet odkazy na umístění požadovaných datasetů. Úložiště pak v definovaném intervalu stáhne datasety podle datového katalogu a uloží je do triplestore databáze.

Dekompozici do modulů je možné vidět na Obr. 6.2. Databázový modul by měl obsluhovat komunikaci s triplestore databází. Konfiguračním modulem je míněno nastavení procesů jednotného úložiště včetně nastavení intervalu stahování datasetů.

Transformační modul

V prvním kroku je třeba načíst jednotlivé datasety subjektů. Odkazy na konkrétní datasety budou reprezentované ve formě datového katalogu. Samotný katalog budeme zapisovat v RDF a serializovat do formátu Turtle. Využijeme k tomu ontologii Data Catalog Vocabulary. Příklad datového katalogu lze vidět v kódu 6.1.

V dalším kroku je třeba vyřešit otázku heterogenity dat. Platforma by principiálně měla umět přijímat RDF data nejen od subjektů zpracovaná konverzním modulem, ale i jakákoli jiná RDF data splňující datový standard a reflektující definovanou RDF ontologii. Jednotlivé entity by ale měly být identifikované podle vzoru z kapitoly Otevřená smlouva jako Linked Data. To nám zaručí, že žádné dvě entity různých subjektů nebudou mít díky rozdílným doménám stejné URI.

Díky tomu tak můžeme v rámci závěrečného kroku data slít dohromady a uložit do triplestore databáze.

```
1 @prefix dcat: <http://www.w3.org/ns/dcat#> .
2 @prefix dct: <http://purl.org/dc/terms/> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5
6 _:dataset a dcat:Dataset ;
7     dcat:distribution _:distribution1 .
8     _:distribution1 a dcat:Distribution ;
9         dct:identifier '1' ;
10        dcat:accessURL <https://raw.githubusercontent.com/
11        PavelHryzlik/ContractStandard/master/standard/lod/
12        contract_sampleA.ttl> ;
13        dct:format [ rdfs:label 'Turtle' ] .
14
15 _:dataset a dcat:Dataset ;
16     dcat:distribution _:distribution2 .
17     _:distribution2 a dcat:Distribution ;
18         dct:identifier '2' ;
19        dcat:accessURL <https://raw.githubusercontent.com/
20        PavelHryzlik/ContractStandard/master/standard/lod/
21        contract_sampleB.ttl> ;
22        dct:format [ rdfs:label 'Turtle' ] .
```

Výpis kódu 6.1: Datový katalog pro jednotné úložiště

Publikační modul

Úložiště by mělo data poskytovat jak k webovému prohlížení, tak skrze API k využití v aplikacích. Tento požadavek vyřešíme vystavením SPARQL endpointu.

Typickou součástí publikace je také registrace datové sady v datovém katalogu.

6.4 Datová síť

Nad jednotným úložištěm může vznikat celá řada aplikací využívajících data o smlouvách. Díky propojení smluv se souvisejícími daty, tak můžeme kontext smluv rozšířit o další informace, resp. demonstrovat výhody principů Linked Data jako datové sítě.

V rámci kapitoly 4 jsme definovali odkaz na veřejnou zakázku (`pco:Contract`) v rámci smlouvy, resp. predikát `pco:publicContract` ze smlouvy (třída `cn:Contract`) a odkaz na ekonomický subjekt (`gr:BusinessEntity`) pomocí predikátů `owl:sameAs` ze Smluvní strany a Vydavatele (třídy `foaf:Organization` a `gr:BusinessEntity`). Nyní si položíme otázku, s jakými dalšími relevantními zdroji můžeme data na základě definovaných odkazů dále propojit.

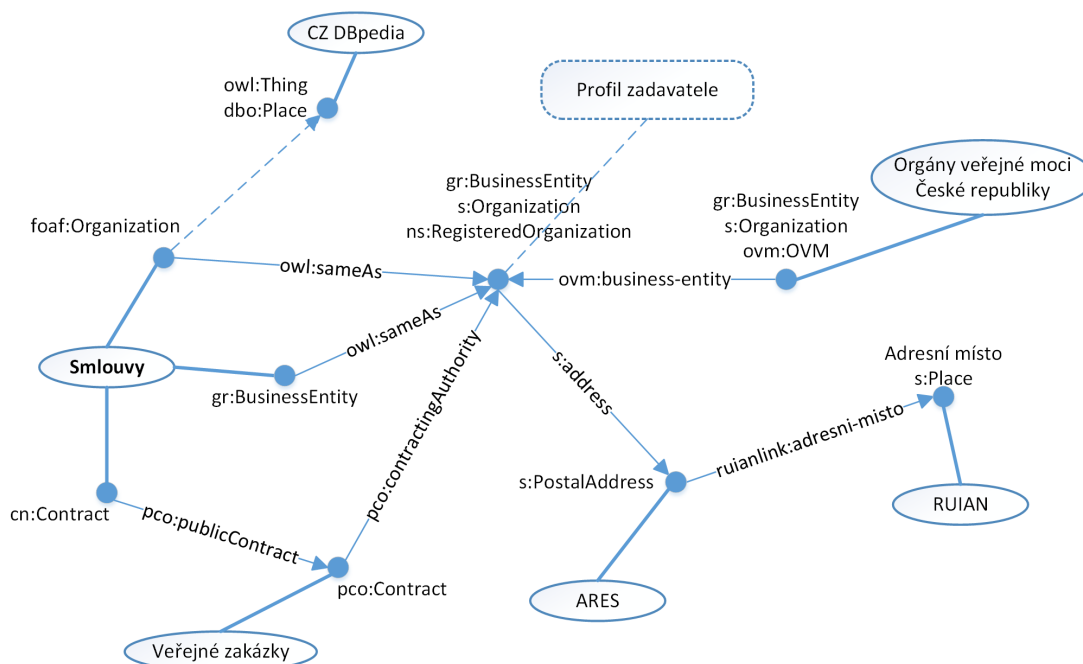
Navrženou datovou síť propojených objektů můžeme vidět na Obr. 6.3. Z každého datasetu (v oválných blocích) vede hrana k vlastnímu objektu. Z nich pak vychází vybrané predikáty odkazující na entitu dalšího datasetu.

Výchozím bodem je entita ekonomického subjektu (`gr:BusinessEntity`) uprostřed. Konkrétně se jedná o registrované organizace, jejichž data vychází z údajů na Profilu zadavatele, z ARESu a dalších zdrojů. Z této entity využijeme predikáty (`s:address`) k propojení s adresou (`s:PostalAddress`) z ARESu. Z adresy, pak získáme informaci o adresním místě (`s:Place`) z RUIANu pomocí predikátu `ruianlink:adresni-misto`.

Z entity ekonomického subjektu můžeme získat informaci o odpovídajícím objektu reprezentovaném v rámci orgánů veřejné moci díky zpětnému odkazu reprezentovanému predikátem `ovm:business-entity`.

Propojení dosáhneme také mezi ekonomickým subjektem a veřejnou zakázkou (`pco:Contract`). Z veřejné zakázky využijeme odkazu na zadavatele, resp. predikátu (`pco:contractAuthority`). Díky tomu můžeme zjistit údaje o veřejných zakázkách jednotlivých ekonomických subjektů, kde vystupují v roli zadavatele.

Vydavatel smluv (`foaf:Organization`) je často veřejná instituce mající svoji stránku na DBpedii. Nedisponujeme však přímým odkazem na konkrétní reprezentaci vydavatele v DBpedii. Můžeme ale položit SPARQL dotaz, zdali v DBpedii existuje instituce s daným konkrétním jménem (porovnáním predikátů `gr:legalName` vydavatele a `rdfs:label` z DBpedie). Nejedná se tedy o přímé propojení, ale o další možnost, jak kontext smluv obohatit o další informace.



Obrázek 6.3: Datová síť

6.5 Webová aplikace

Úkolem webové aplikace je nad jednotným úložištěm zpřístupnit údaje o smlouvách k prohlížení koncovým uživatelům. Řešení rozdělíme do čtyř modulů (viz Obr. 6.2).

Endpoint modul

V rámci tohoto modulu definujeme napojení na požadované zdroje dat v podobě SPARQL endpointů.

Procesní modul

Úkolem tohoto modulu je načíst požadované údaje o smlouvách a přiřadit k nim údaje z rozšířeného kontextu navrženého v minulé kapitole. Cílem je tedy získat a sjednotit informace z několika zdrojů dat.² Existuje několik přístupů, jak se nad požadovanými daty dotazovat³ - z klientské části aplikace, nebo ze serverové. Z důvodu dalšího zpracování dat se jeví vhodnější serverové dotazování. Samotné dotazy můžeme pokládat také různými způsoby. Jednou z možností je distribuované dotazování, kdy se v rámci jednoho dotazu můžeme odkazovat na více zdrojů. Další možností je se nad každým zdrojem dotazovat zvlášť, resp. sadou dotazů. V souladu s principy Linked Data zvolíme sadu dotazů nad jednotlivými SPARQL endpointy. Výhodou je, že problém jednoho zdroje by neměl ovlivnit výsledky z ostatních zdrojů.

²Ukázka skici obohaceného kontextu smluvních údajů viz Obr. 6.4

³Předpokládáme, že aplikace bude obsahovat serverovou část.

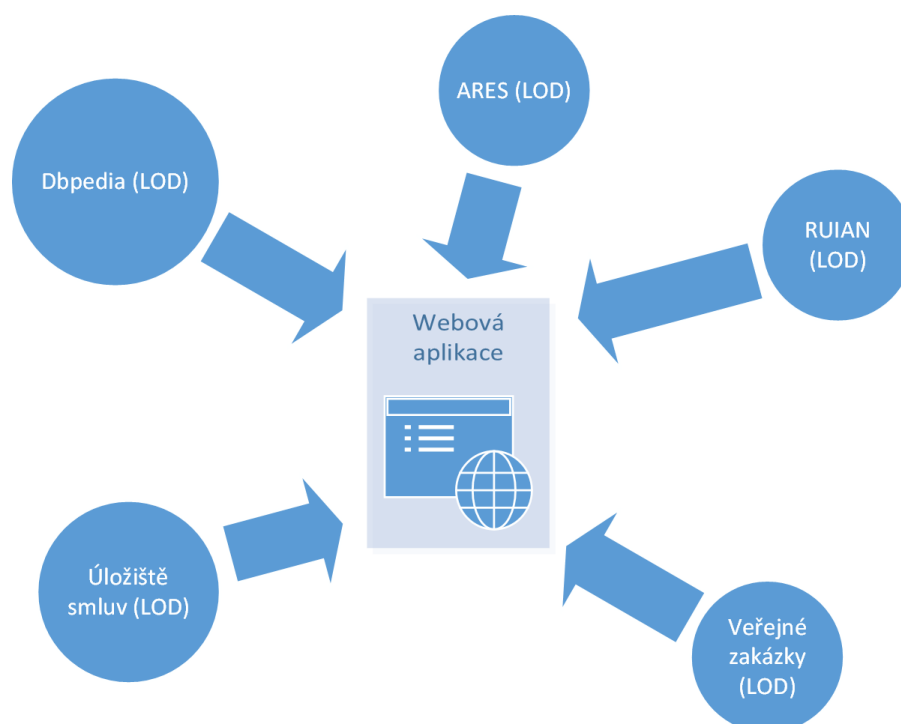
Nad SPARQL endpointy můžeme získávat data buď ve formě RDF (příkaz Construct), nebo v tabulkové formě (příkaz Select). Účelem je zobrazení dat uživateli často právě ve formě tabulek. Volba příkazu Select je tedy lepší volbou.

Komunikační modul

Agendou komunikačního modulu je výměna dat mezi procesním a prezentačním modulem. Modul zpracuje požadavky z prezentačního modulu a volá konkrétní funkce procesního modulu. Výsledná data pak pošle prezentačnímu modulu k zobrazení. Rozhraní pro přenos by mělo být ve standardizovaném datovém formátu⁴.

Prezentační modul

Účelem tohoto modulu je zobrazování dat uživateli. Výchozím bodem je nabídnout přehledné informace o otevřených smlouvách. Úvodní obrazovku bude tvořit pohled vyznačující subjekty otevírající smlouvy (ideálně na mapovém podkladu) a následně souhrnný seznam všech smluv. Z informací o subjektech půjde přejít na detail vybraného subjektu. Dále z každé položky seznamu půjde přejít jak do detailu konkrétní smlouvy, tak do detailu jejího vydavatele (subjektu). Detail smlouvy zobrazí podrobné informace o zvolené smlouvě, včetně jejích verzí, smluvních stran, příloh, dodatků a milníků. Ze subjektů a smluvních stran majících vyplněný IČ půjde přejít na seznam veřejných zakázek, ve kterých subjekt vystupuje. Detail vydavatele nabídne podrobnější informace o zvoleném publikujícím subjektu. Součástí detailu vydavatele bude také seznam jeho smluv. Z tohoto seznamu půjde také přejít na detaily jednotlivých smluv.



Obrázek 6.4: Obohacený kontext smluv díky propojeným datům

⁴Typicky v XML, nebo JSON

7. Implementace platformy

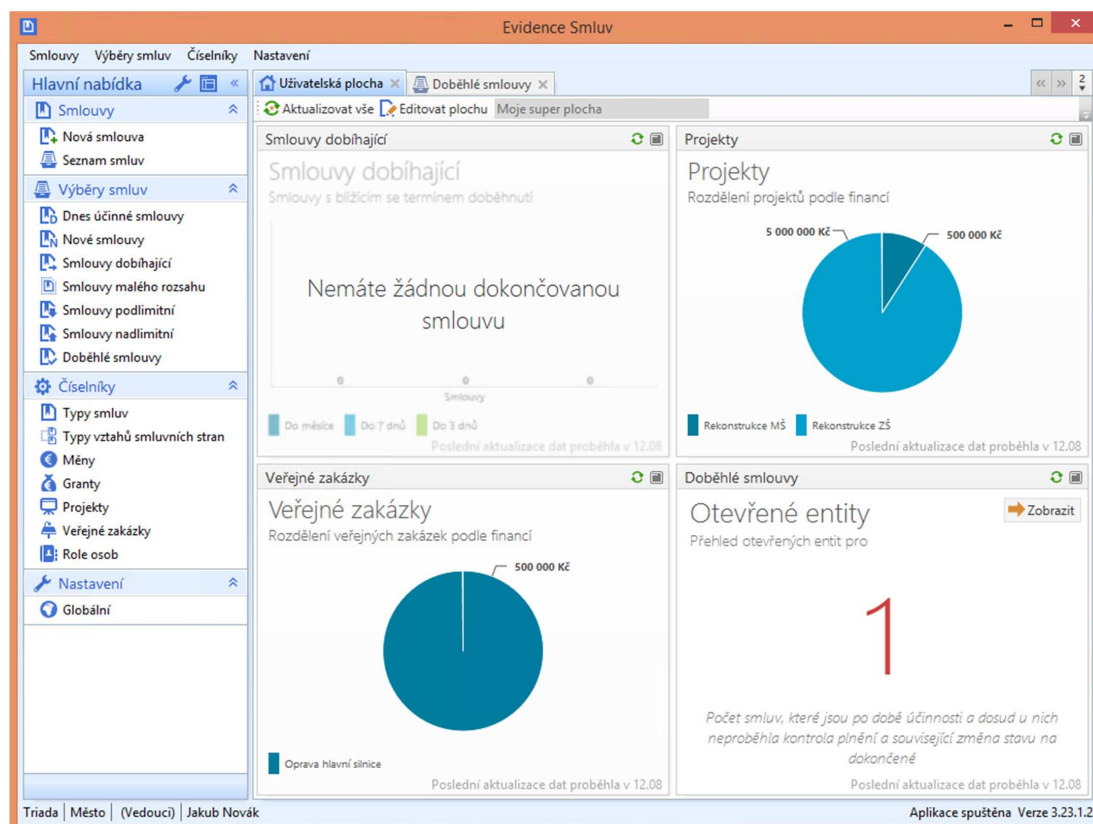
7.1 Konverzní mechanismus

Prvním krokem ve vývoji konverzního modulu je volba a analýza zdrojové platformy. Je třeba zpracovat danou doménu a především se seznámit se strukturou datového modelu. Druhým krokem je tvorba R2RML mapovacího skriptu. Závěrečným krokem je samotná implementace konverzního mechanismu. Tento postup uvádíme proto, že jednotlivé kroky lze řešit nezávisle. Např. definované R2RML mapování nad konkrétní doménou může posloužit jako podklad k různým aplikacím pracujícím s R2RML.

7.1.1 Munis ESML

Jako zdroj pro implementaci konverzního mechanismu byl zvolen modul Munis ESML. Jedná se o část informačního systému pro města a obce společnosti Triada spol. s.r.o.

Účelem modulu Munis ESML je evidování odběratelských i dodavatelských smluv. Nabízí přehledné vyhledávání, statistiky, hlídání termínů, nebo možnost přiřadit smlouvy jednotlivým grantům, projektům či veřejným zakázkám (viz Obr. 7.1).



Obrázek 7.1: Modul ESML

7.1.1.1 Struktura datového modelu

Základem datového modelu jsou entity *Smlouva* a *Verze smlouvy*. Smlouva je základním stavovým objektem s hierarchickou strukturou. Vycházíme z předpokladu, že dodatek ke smlouvě je také smlouva, proto definujeme:

- Entita *Smlouva* na kořenové úrovni popisuje smlouvu
- Každý syn entity *Smlouva* je jejím dodatkem

Každá smlouva je verzovaná, resp. entita *Smlouva* může mít několik *Verzí smlouvy*. Entita *Verze smlouvy* reprezentuje popisné údaje *smlouvy*. Dále obsahuje vazby na *rozdělovník*, *smluvní strany*, *mílníky*, *transakce*, *externí kontakty* a *číselníky*, viz Obr. 7.2.

Každá *Verze smlouvy* může obsahovat hierarchickou strukturu příloh. Každá entita *Příloha smlouvy* reprezentuje fyzický soubor. Přílohy definujeme takto:

- Každá *Verze smlouvy* může mít pouze jednu kořenovou přílohu
- Kořenová příloha je hlavním dokumentem obsahujícím text smlouvy
- Ostatní jsou dílčími přílohami

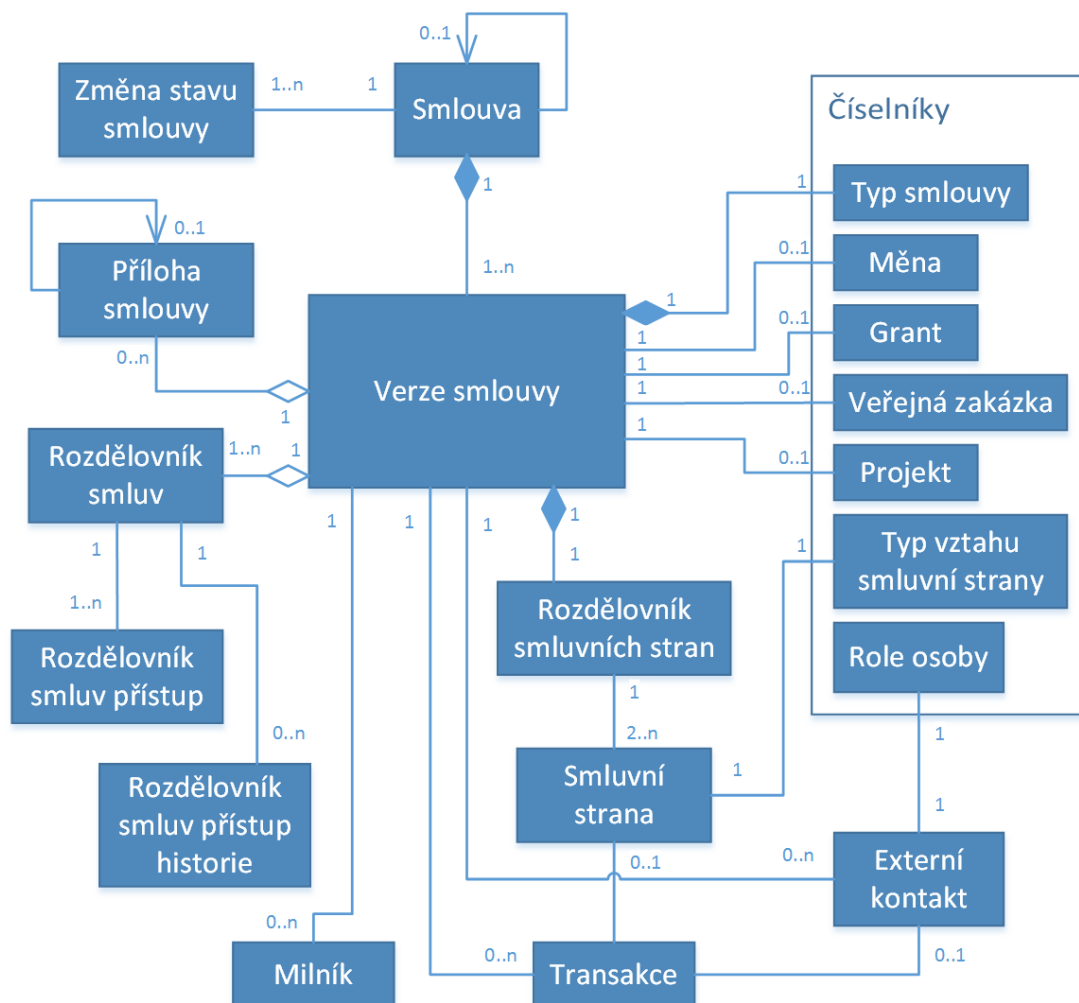
Entity *Změna stavu smlouvy*, *Rozdělovník*, *Rozdělovník smluv přístup*, *Rozdělovník smluv přístup historie* nejsou pro naše účely důležité, proto je dále v textu nebudeme zmiňovat.

K popsání informací o smlouvách budeme ještě využívat entitu *tri_uzivatel* reprezentující uživatele systému a *tri_orgadr* reprezentující adresu útvaru.

7.1.1.2 Omezení vůči standardu

V porovnání s datovým standardem pro smlouvy disponuje Munis ESML několika omezeními:

- Modul nepodporuje objednávky a faktury
- Transakce nejsou implementovány - s podporou transakcí a obecně smluvního plnění se počítá do dalších verzí



Obrázek 7.2: Zjednodušený datový model (bez atributů) Munis ESML

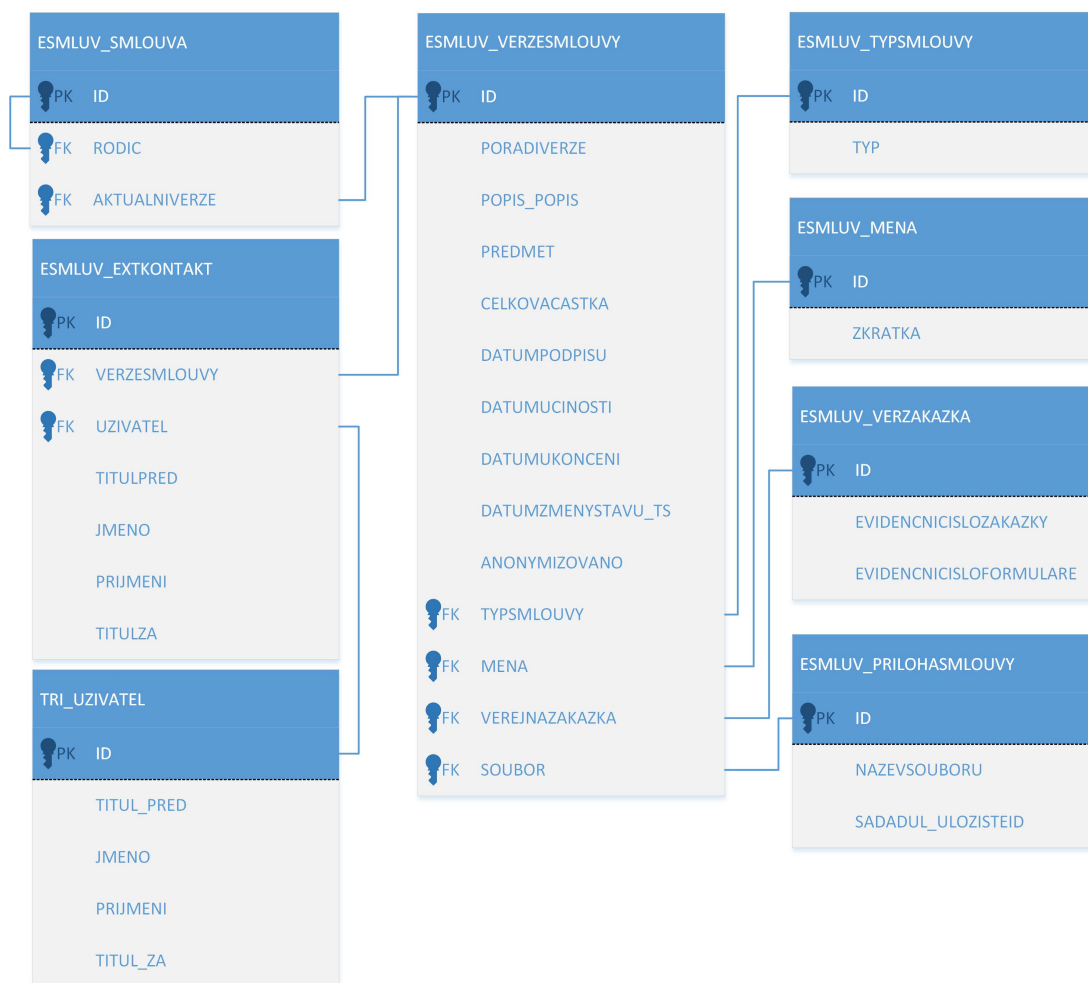
7.1.2 R2RML mapování

Pomocí R2RML skriptu můžeme namapovat konkrétní sloupce z databázových tabulek na RDF predikáty. Pro složitější mapování umožňuje R2RML definovat vlastní SQL pohledy (SQL Views) nad relační databází, čehož využijeme. Pro každou entitu v rámci datového standardu proto definujeme vlastní SQL pohled. Výsledný R2RML skript lze nalézt v příloze E.

V následující části je schématicky naznačeno mapování položek. Každé entitě přiřadíme URI a typ. Následně se namapují jednotlivé položky na predikáty. Vycházíme z informací řečených v kapitole Otevřené smlouvy jako Linked Data.

7.1.2.1 Smlouva

Na obr. 7.3 můžeme vidět vybraný seznam tabulek a jejich atributů z datového modelu Munis ESML, které použijeme k mapování smlouvy.



Obrázek 7.3: R2RML mapování - tabulky k mapování smlouvy

Prvním krokem je přiřazení URI entitě. URI je kombinací atributu ID tabulky **ESMLUV_SMLOUVA** a číslem její verze, resp. atributem **PORADIVERZE** tabulky (**ESMLUV_VERZESMLOUVY**). Typem entity je třída **cn:Contract**.

- **URI entity** - `http://[domain]/contract/{ID}/{PORADIVERZE}`
- **Typ** - **cn:Contract**

Konstanty Některé položky budou vždy obsahovat konstantní hodnotu:

- **dcmi:type** - s hodnotou „Smlouva“
- **cn:priceAnnual** - Nelze určit roční částku, proto vždy „false“
- **cn:funding** - Vychází zatím z nedefinovaného číselníku datového standardu, mapujeme nepovinně konstantní hodnotu „vlastní“

Mapované položky

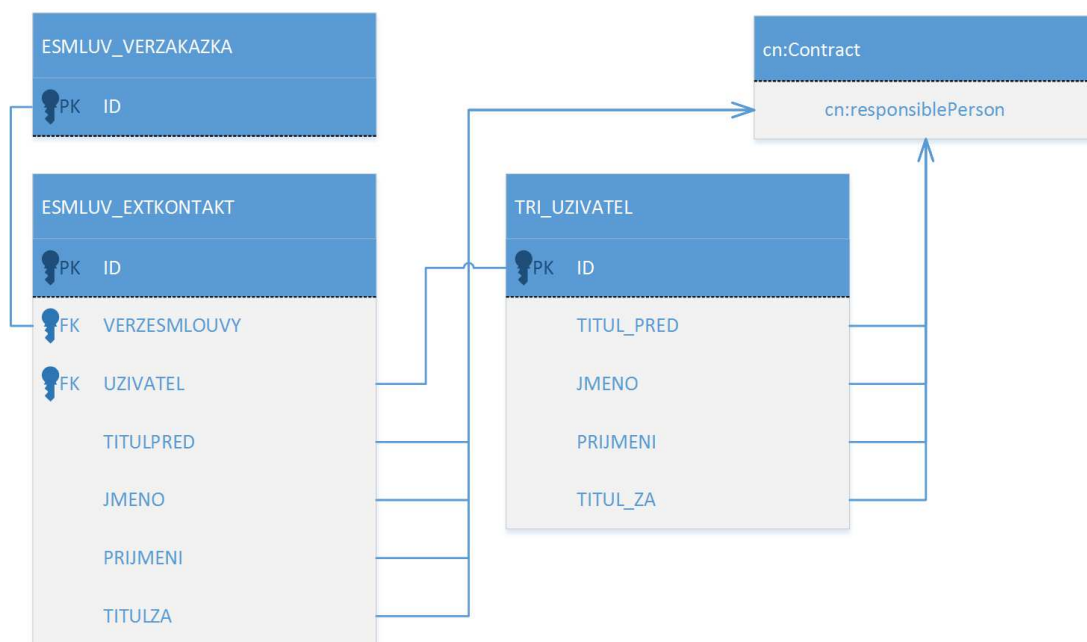
- **cn:anonymised** - Atribut **ANONYMIZOVANO**
- **dc:title** - Atribut **PREDMET**

- **dc:description** - Atribut POPIS_POPIS
- **dc:created** - Atribut DATUMPODPISU
- **cn:validFrom** - Atribut DATUMUCINOSTI
- **cn:validUntil** - Atribut DATUMUKONCENI
- **cn:valid** - Položka je „true“, jestliže se jedná o nejnovější verzi smlouvy, jinak je „false“
- **cn:contractType** - Atribut TYP, mapován na odpovídající hodnotu číselníku typů smlouvy
- **cn:competency** - Vyplní se na základě položky TYP. Pokud je Typ smlouvy - „Veřejnoprávní smlouva“ vyplní se i do položky cn:competency, jinak se vyplní „Soukromoprávní smlouva“
- **cn:awardID** - Atribut EVIDENCNICISLOZAKAZKY
- **cn:awardProfileID** - Atribut EVIDENCNICISLOFORMULARE
- **pc:publicContract** - Atribut EVIDENCNICISLOZAKAZKY ve formě odkazu na věstník veřejných zakázek (LinkedData podoba)
 - *http://linked.opendata.cz/resource/domain/buyer-profiles/-contract/cz/{EVIDENCNICISLOZAKAZKY}*
- **cn:amount** - Odkaz na podrobné informace o ceně
 - *http://[domain]/contract/{ID}/{PORADIVERZE}/amount*
 - Mapování probíhá na objekt typu **gr:PriceSpecification** s výše zmíněným URI. Namapovány jsou položky:
 - * **gr:hasCurrencyValue** - Atribut CELKOVACASTKA
 - * **gr:hasCurrency** - Atribut ZKRATKA
- **dc:publisher** - Odkaz na vydavatele
 - *http://[domain]/publisher*
- **cn:version** - Odkaz na informace o verzi smlouvy
 - *http://[domain]/contract/{ID}/{PORADIVERZE}/version*
- **schema:url** - Odkaz na fyzický dokument smlouvy
 - *http://[domain]/file/{SADADUL_ULozISTEID}/{NAZEVSouboru}*
- **cn:party** Odkaz na smluvní strany. Mapování probíhá skrze tabulku ESMLUV_SMLSTRANROZD na které odkazují jak verze smlouvy, tak smluvní strana. Více viz R2RML skript v příloze E.
 - *http://[domain]/party/{HAD_POUZITA}*

- **cn:responsiblePerson** - Každá veřejná zakázka má vazbu na externí kontakty. Externím kontaktem může být buď uživatel informačního systému (tabulka TRI_UZIVATEL), nebo jakákoli osoba vyplněná v tabulce Externí kontakt. Pro potřeby mapování se hodnoty spojí do jednoho řetězce, viz Obr. 7.4.
- **cn:amendment** Odkaz na dodatky. Dodatek v Munis ESML je také smlouvou, ale vždy synem jiné smlouvy nebo dodatku. Smlouvě tedy přiřadíme ty dodatky, jejichž atribut RODIC a PORADIVERZE odpovídají dané smlouvě.
 - *http://[domain]/amendment/{ID}/{PORADIVERZE}*
- **cn:attachment** Odkaz na přílohy. Každá příloha má odkaz na smlouvu (viz Obr. 7.6)
 - *http://[domain]/attachment/{ID}/1*
- **cn:implementation** Odkaz na objekt Implementace
 - *http://[domain]/contract/{ID}/{PORADIVERZE}/implementation*

Nenamapované položky

- **cn:uri** - Položka odpovídá URI entity
- **cn:amountNoVat** - Cena bez dph, předpokládaná podpora spolu s podporou podrobného smluvního plnění
- **cn:subjectType** - Číselník typů zboží/slужeb, předpokládaná podpora u dalších verzí
- **cn:plainText** - Prostý text dokumentu smlouvy, resp. alternativa k oskenovaným dokumentům. Vyžaduje hlubší analýzu procesu zpracování dokumentů



Obrázek 7.4: R2RML mapování externího kontaktu

7.1.2.2 Smluvní strana

Mapování smluvní strany a adresy vychází z údajů tabulek ESMLUV_SMLUVSTRANA a HAD_POUZITA (viz Obr. 7.5)

- **URI** - *http://[domain]/party/{HAD_POUZITA}*
- **Typ** - *gr:BusinessEntity*

Mapované položky

- **gr:legalName** - Atribut NAZEV_SUBJEKTU
- **dc:identifier** - Atribut ICO
- **owl:sameAs** - Atribut ICO, odkaz na reprezentaci ekonomického subjektu v ARESu (LinkedData podoba)
 - *http://linked.opendata.cz/resource/business-entity/CZ{ICO}*
- **cn:noID** - „true“ pokud je vyplněno IČ, jinak „false“
- **cn:localID** - Atribut HAD_POUZITA
- **schema:addressCountry** - Atribut STAT
- **cn:paysVAT** - Atribut PLATCEDPH
- **schema:address** - Odkaz na adresu
 - *http://[domain]/party/{HAD_POUZITA}/address*

Nenamapované položky

- **cn:payer** - Modul ESML zatím neviduje kompletní smluvní plnění, takže nelze určit
- **cn:superiorInstitution** - Modul ESML neviduje nadřazené instituce

7.1.2.3 Adresa

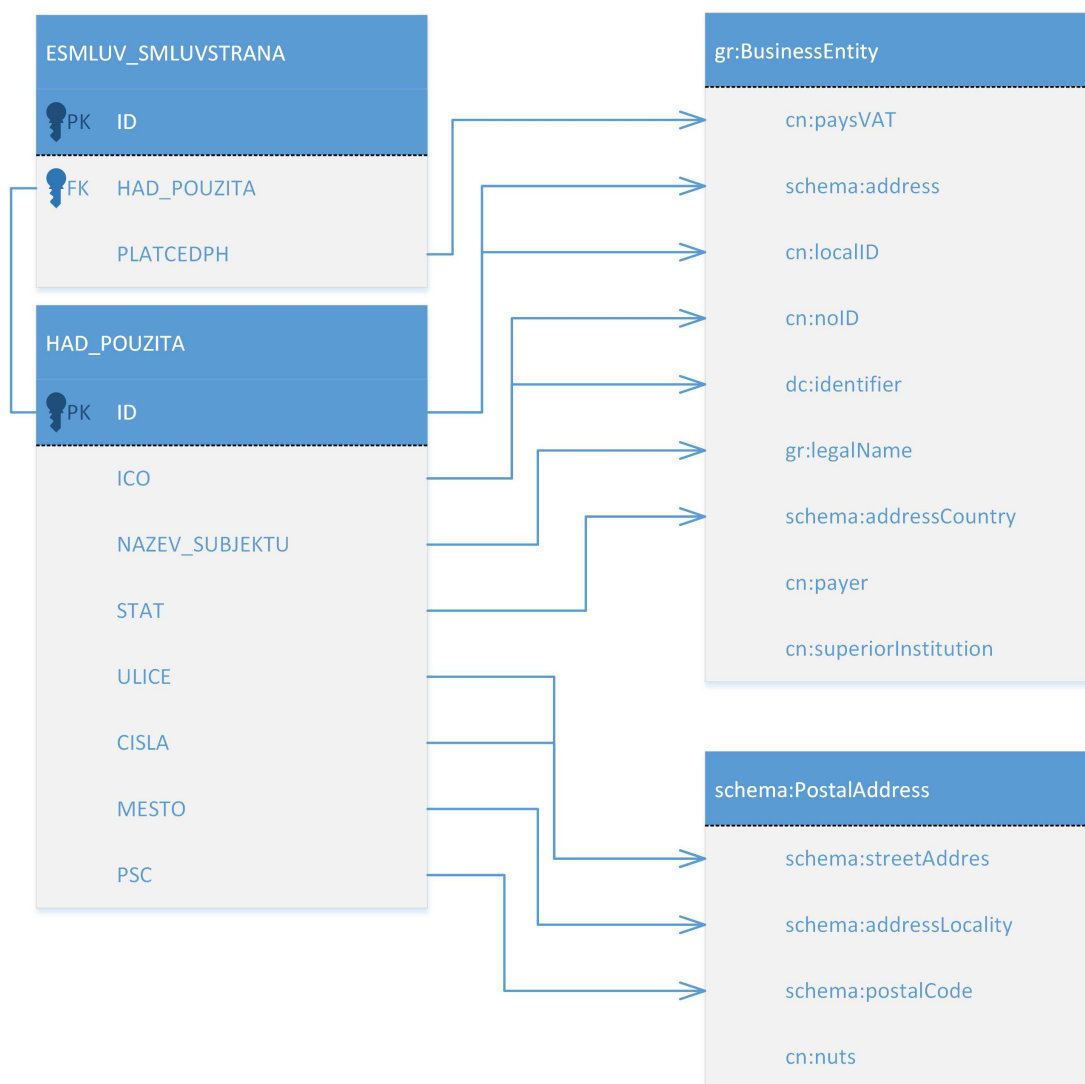
- **URI** - *http://[domain]/party/{HAD_POUZITA}/address*
- **Typ** - schema:PostalAddress

Mapované položky

- **schema:streetAddress** - Složení atributu ULICE a CISLA
- **schema:postalCode** - Atribut PSC
- **schema:addressLocality** - Atribut MESTO

Nenamapované položky

- **cn:nuts** - Modul ESML neviduje hodnoty normalizované klasifikace územních celků



Obrázek 7.5: R2RML mapování vlastností Smluvní strany a Adresy

7.1.2.4 Příloha

Mapování přílohy vychází z údajů tabule ESMLUV_PRILOHASMLOUVY (viz Obr. 7.6). V modulu smluv lze přílohy editovat, ale neprovádí se jejich verzování. Proto verze přílohy bude mít vždy hodnotu „1“. Oproti datovému standardu, příloha v datovém modelu Munis ESML popisuje spíše fyzický soubor, nemůžeme tedy namapovat většinu položek entity Dokument, ale jen některé. Nemapované položky proto neuvádíme.

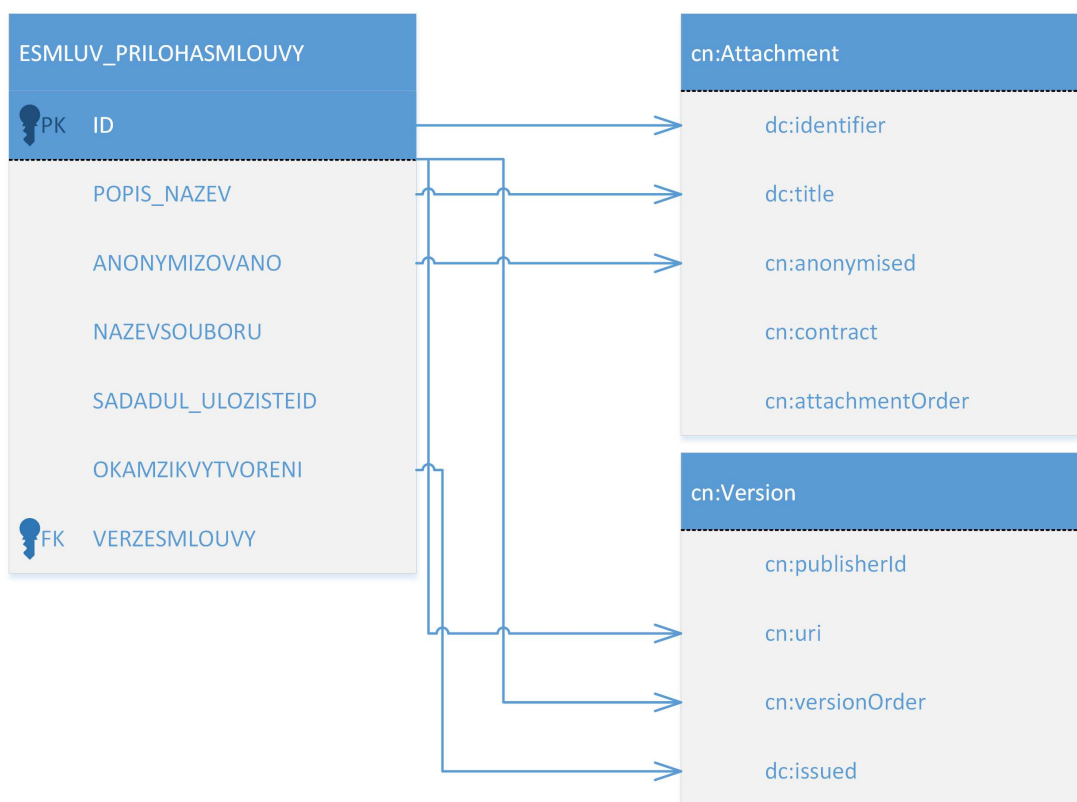
- **URI** entity - *http://[domain]/attachment/{ID}/1*
- **Typ** - cn:Attachment

Konstanty

- **dcmi:type** - Hodnota „Příloha“
- **cn:valid** - Položka je vždy „true“, přílohy nejsou verzované

Mapované položky

- **cn:anonymised** - Atribut ANONYMIZOVANO
- **dc:title** - Atribut POPIS_NAZEV
- **dc:identifier** - Atribut ID
- **schema:url** - Odkaz na fyzické umístění souboru
 - *http://[domain]/file/{SADADUL_ULOZISTEID}/{NAZEVSOUBORU}*
- **dc:publisher** - Odkaz na vydavatele
 - *http://[domain]/publisher*
- **cn:version** - Odkaz na informace o verzi smlouvy
 - *http://[domain]/attachment/ID/1/version*
- **cn:contract** - Odkaz na nadřizenou smlouvu. K tomuto údaji je třeba ID Smlouvy (tabulka ESMLUV_SMLOUVA a pořadí verze smlouvy, tabulka ESMLUV_VERZESMLOUVY)
 - *http://[domain]/attachment/{ID}/{PORADIVERZE}*



Obrázek 7.6: R2RML mapování vlastností Příloha

7.1.2.5 Dodatek

Dodatek v datovém modelu Munis ESML je také smlouvou, takže budeme vycházet z atributů jako u smlouvy (viz Obr. 7.3).

- **URI entity** - *http://[domain]/amendment/{ID}/{PoradiVerzeDodatku}*
- **Typ** - cn:Amendment

Konstanty

- **dcmi:type** - s hodnotou „Dodatek“

Mapované položky

- **cn:anonymised** - Atribut ANONYMIZOVANO
- **dc:title** - Atribut PREDMET
- **dc:identifier** - Atribut ID
- **dc:created** - Atribut DATUMPODPISU
- **cn:valid** - Položka je „true“, jestliže se jedná o nejnovější verzi dodatku, jinak je „false“
- **cn:contract** - Odkaz rodičovskou smlouvu
 - *http://[domain]/contract/{RODIC}/{PORADIVERZE}*
- **dc:publisher** - Odkaz na vydavatele
 - *http://[domain]/publisher*
- **cn:version** - Odkaz na informace o verzi dodatku
 - *http://[domain]/amendment/{ID}/{PORADIVERZE}/version*
- **schema:url** - Odkaz na fyzický dokument dodatku
 - *http://[domain]/file/{SADADUL_ULozISTEID}/{NAZEVSOUBORU}*
- **cn:responsiblePerson** - Každá veřejná zakázka má vazbu na externí kontakty. Externím kontaktem může být buď uživatel informačního systému (tabulka TRI_UZIVATEL), nebo jakákoli osoba vyplněná v tabulce Externí kontakt. Pro potřeby mapování se hodnoty spojí do jednoho řetězce, viz Obr. 7.4.

Nenamapované položky

- **cn:uri** - Položka odpovídá URI entity
- **cn:plainText** - Prostý text dokumentu smlouvy, resp. alternativa k oskenovaným dokumentům. Vyžaduje hlubší analýzu procesu zpracování dokumentů

7.1.2.6 Verze dokumentu

K mapování jednotlivých verzí smlouvy/přílohy/dodatku. Pro smlouvy/dodatky mapujeme z tabulky ESMLUV_VERZESMLOUVY (viz Obr. 7.7). Pro přílohu mapujeme z tabulky ESMLUV_PRILOHA (viz. Obr. 7.6)

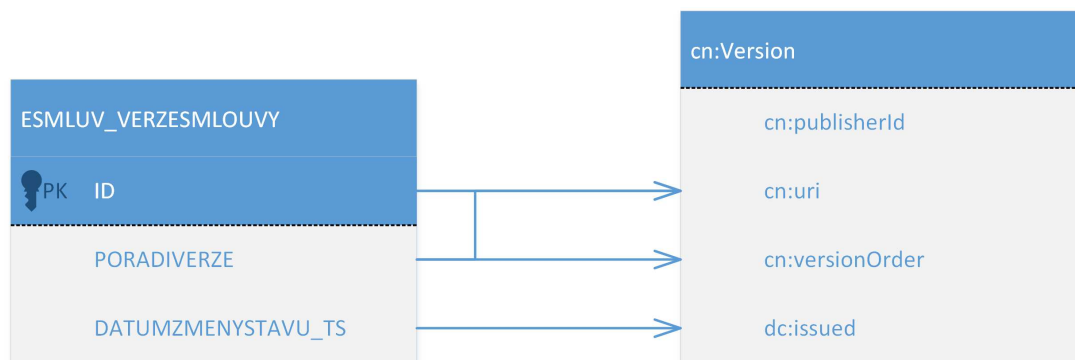
- **URI entity**
 - U smlouvy/dodatku -
http://[domain]/[type]/{ID}/{PORADIVERZE}/version
 - U přílohy - *http://[domain]/attachment/{ID}/1/version*
- **Typ** - cn:Version

Mapované položky

- **cn:uri** - Položka je stejná jako URI entity
- **cn:versionOrder** - U smlouvy/dodatku atribut PORADIVERZE, u přílohy hodnota „1“
- **dc:issued** - U smlouvy/dodatku atribut DATUMZMENYSTAVU_TS, u přílohy hodnota OKAMZIKVYTVORENI

Nenamapované položky

- **cn:publisherId** - Díky Id a verzi dokumentu máme každou entitu jednoznačně identifikovanou, proto není třeba vyplňovat.



Obrázek 7.7: R2RML mapování vlastností Verze

7.1.2.7 Implementace

Entitu implementace nereprezentuje žádná tabulka v rámci datového modulu ERMS. Je třeba ji proto vytvořit s odpovídajícími položkami. Využijeme tabulek ESMLUV_SMLOUVA, ESMLUV_VERZESMLOUVY a ESMLUV_MILNIK.

- **URI entity** - *http://[domain]/contract/ID/PORADIVERZE/implementation*
- **Typ** - cn:Implementation

Mapované položky

- **cn:milestone** - Položka je stejná jako URI entity
 - textithttp://[domain]/contract/{ID}/{PORADIVERZE}/milestone/{MilestoneID}

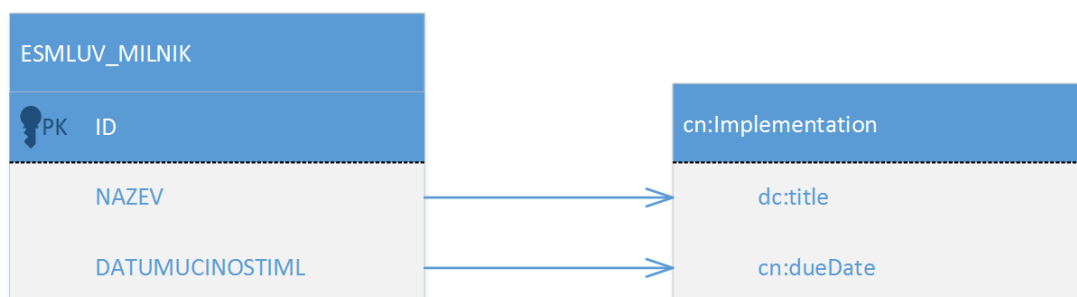
7.1.2.8 Milník

K mapování milníků využijeme tabulek ESMLUV_SMLOUVA, ESMLUV_VERZESMLOUVY a ESMLUV_MILNIK (tabulka milníku viz Obr. 7.8).

- **URI entity** - *http://[domain]/contract/{ID}/{PORADIVERZE}/milestone/{MilestoneID}*
- **Typ** - cn:Milestone

Mapované položky

- **dc:title** - Atribut NAZEV
- **cn:dueDate** - Atribut DATUMUCINOSTIML



Obrázek 7.8: R2RML mapování vlastností Milníku

7.1.2.9 Vydavatel

Vydavatele namapujeme pomocí tabulky TRI_ORGADR, viz Obr. 7.9

- **URI entity** - *http://[domain]/publisher*
- **Typ** - foaf:Organization

Konstanty

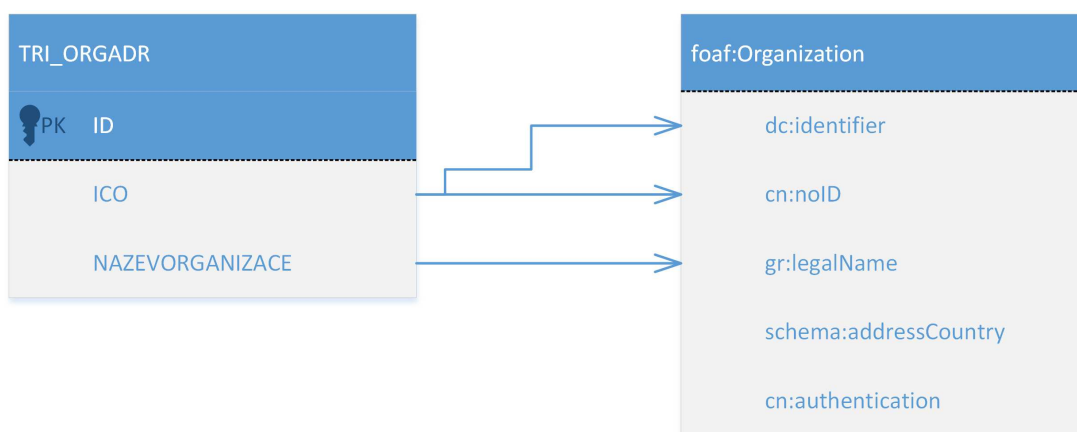
- **schema:addressCountry** - Hodnota „CZE“

Mapované položky

- **gr:legalName** - Atribut NAZEVORGANIZACE
- **cn:noID** - „true“ pokud je vyplněno IČ, jinak „false“
- **dc:identifier** - Atribut ICO
- **owl:sameAs** - Atribut ICO, odkaz na reprezentaci ekonomického subjektu v ARESu (LinkedData podoba)
 - *http://linked.opendata.cz/resource/business-entity/CZ{ICO}*

Nenamapované položky

- **cn:authentication** - Pro naše účely nemá smysl



Obrázek 7.9: R2RML mapování vlastností Vydavatele

7.1.3 Volba technologií a implementační platformy

Pro samotnou implementaci konverzního mechanismu zvolíme platformu .Net. Modul bude mít formu webové aplikace, resp. virtuálního SPARQL endpointu, kterou budeme implementovat v technologii ASP.Net. Využijeme tradičního architektonického vzoru MVC. K práci s RDF daty budeme využívat knihovnu dotnetRdf.

7.1.3.1 Volba R2RML procesoru

K R2RML mapování využijeme projektu DotNetR2RMLStore[58][59]. Jedná se o experimentální R2RML procesor pracující nad relačními databázemi Microsoft SQL¹. Tento projekt je vytvářený v rámci Katedry softwarového inženýrství na Matematicko-fyzikální fakultě.

¹Většina veřejných institucí využívající produkty firmy Triada. spol, s.r.o. pracuje nad databázemi MS SQL. Proto nebereme MS SQL jako omezení. Munis ESML ale umožňuje práci i nad databází Oracle.

Omezení R2RML procesoru

Využití zmíněného R2RML procesoru si vyžádalo několik drobných omezení.

Pro náš případ využijeme DotNetR2RMLStore ve verzi 0.0.0.9. Zkoušená vyšší verze mění logiku zpracování dotazů a zatím nepodporuje SQL Views. Je tu i možnost vytvořit SQL views přímo nad databází subjektu a mapovat R2RML skriptem přímo tyto pohledy. Není to problém, ale nelze považovat za samozřejmost, že subjekt zpřístupní databázi k úpravám.

Procesor nepodporuje SPARQL příkazy ASK a DESCRIBE. Pro naše účely ale stačí hlavní příkazy SELECT a CONSTRUCT.

Samotný R2RML skript musí mít v hodnotě *template* vždy vyplněné absolutní URI. Prefixovaný zápis procesor zpracuje, ale při zpracování dotazů daný template nerozpozná.

Formáty datumových položek v RDF datech by měly splňovat W3C specifikaci, což aktuálně nesplňují. Vrácené datumové hodnoty tedy v rámci postprocessingu nahradíme správným formátem[60].

7.1.3.2 Napojení na datové úložiště

Mezi specifika informačních systémů firmy Triada s.r.o můžeme zmínit, že neukládají fyzické soubory (v našem případě smlouvy) do databáze s ostatními daty, ale do specializovaného datového úložiště. Nutnou podmínkou pro zobrazení těchto dat je proto propojení konverzního modulu s databází datového úložiště. Využijeme k tomu knihovnu TriadaModulZaklad.

V relační databázi jsou uloženy informace o daném souboru. Jedná se mimo jiné o název souboru a jeho jednoznačný indentifikátor v datovém úložišti ve formě GUID. Informace tedy namapujeme již na zmíněné URI - *http://[domain]/file/SADADUL_ULOZISTEID/NAZEVSOUBORU*. Při přístupu na danou adresu se informace převedou na dotaz do datového úložiště a uživateli se vrátí konkrétní soubor ke stažení.

7.1.4 SPARQL endpoint

Virtuální SPARQL endpoint vystavený nad konverzním modulem je dostupný na adrese:

- *http://[domain]/sparql*

Interface endpointu se skládá z jednoduchého pohledu obsahující formulář pro zadávání SPARQL dotazů a volbu požadovaného výstupního formátu. Každý zadaný SPARQL dotaz je enkódován jako HTTP Get na základě kterého pak modul vrátí požadovaná data. Obdobně také probíhá dereferencování entit. Zadané HTTP URI reprezentující konkrétní entitu se převede na odpovídající SPARQL příkaz CONSTRUCT. Výsledný dotaz pak vypadá takto:

- *http://[domain]/sparql?query={SPARQLQUERY}&Format={OUTPUT}*
- SPARQLQUERY značí SPARQL dotaz

- OUTPUT reprezentuje požadovaný výstupní formát². Defaultní hodnotou je formát HTML.

Možnost DUMPu dat v podstatě znamená příkaz CONSTRUCT nad všemi daty. Má však speciální konstrukci:

- `http://[domain]/dump?Format={OUTPUT}&Store={STORE}`
- OUTPUT reprezentuje požadovaný výstupní formát
- STORE nabývá buď hodnoty InMemory (zpracování v paměti), nebo Stream (proudové zpracování). Defaultní hodnotou je InMemory.

7.1.5 Zpracování RDF výstupu

Příkazy SELECT jsou zpracovávány proudově v tabulkové formě, resp. seznamem definovaných proměnných a výčtem hodnot, které jim odpovídají. Definujeme proto handler naslouchající nad R2RML procesorem, kterým výsledky dotazu postupně zpracováváme. Pro každý výstupní formát proto implementujeme handler serializující výsledky do zvoleného datového formátu. Aplikace podporuje základní formáty jako HTML, Turtle, N-Triples, RDF/XML, XML, JSON a CSV. Formáty XML, JSON, CSV serializujeme jako SPARQLResults podle doporučení W3C[61].

Výhodou proudového přístupu je možnost zpracování teoreticky neomezeného množství dat s dobou zpracování lineárně závisující na daném vstupu.

Příkazem CONSTRUCT získáme na výstupu RDF graf ve formě trojic. Dotaz můžeme zpracovávat jak proudově, tak v paměti. Nevýhodou proudového zpracování je, že nám výsledky přicházejí postupně, data proto lze jen obtížně zkracovat pomocí prefixů, sdružovat související informace apod. V druhém případě máme výsledek uložený v interní reprezentaci jako RDF graf. Graf tedy můžeme procházet a formátovat libovolným způsobem. Nevýhodou jsou vysoké paměťové nároky. Aplikace v obou případech podporuje serializaci do formátů HTML, Turtle, N-Triples, RDF/XML, XML, JSON a CSV. Výstup HTML také slouží k prohlížení dat. Jednotlivé URI jsou ve formě hypertextových odkazů, lze tedy procházet mezi provázanými entitami.

Zpracování JSON-LD

Zvláštní kapitolou je formát JSON-LD. Tento formát není určen pro dotazování, ale spíše na zpracování výsledného grafu. Zavedeme ho tedy jako další možnost zpracování DUMPu dat. Ke zpracování RDF dat potřebujeme načíst definovaný JSON-LD Context, provést mapování nad RDF daty a následně strukturu upravit tak, aby byla validní vůči JSON schématu datového standardu. K mapování využijeme knihovnu JSON-LD.Net. Knihovna však nereflktuje JSON datové typy, resp. všechny hodnotové typy jsou String. Výsledek by tak nebyl validní vůči JSON schématu. Lehce tedy knihovnu upravíme, aby vracela požadované datové typy (viz Kód 7.1).

²U derefencovaných entit je vždy výstupní formátem HTML. Je to z důvodu možnosti prohlížení a listování mezi entitami pomocí hypertextových odkazů, viz Zpracování RDF výstupu.

```

1 // Convert value to corresponding type (reflects JSON Schema types)
2 // e.g. DateTime is string in JSON Schema with dateTime formatting
3 switch (((JValue) value["@type"]).Value.ToString())
4 {
5     // JSON Schema type – boolean
6     case XmlSpecsHelper.XmlSchemaDataTypeBoolean:
7         bool boolValue;
8         if (bool.TryParse(stringValue, out boolValue))
9         {
10            return new JValue(boolValue);
11        }
12        break;
13    // JSON Schema type – integer
14    case XmlSpecsHelper.XmlSchemaDataTypeInteger:
15        int integerValue;
16        if (int.TryParse(stringValue, out integerValue))
17        {
18            return new JValue(integerValue);
19        }
20        break;
21    // JSON Schema type – number
22    case XmlSpecsHelper.XmlSchemaDataTypeFloat:
23        float floatValue;
24        if (float.TryParse(stringValue, out floatValue))
25        {
26            return new JValue(floatValue);
27        }
28        break;
29 }
30 return new JValue(stringValue);

```

Výpis kódu 7.1: Rozšíření knihovny JSON-LD.Net

7.1.6 Konfigurace

Veškeré důležité nastavení aplikace se nachází v souboru *Web.config*, převážně:

- Nastavení *ConnectionStringu* k relační databázi
- Nastavení přístupových údajů k datovému úložišti firmy Triada s.r.o

Samotný R2RML mapovací skript je umístěn ve složce *App_Data* v kořenové větvi projektu.

7.1.7 Požadavky na architekturu

R2RML procesor zajišťuje komunikaci s databází a samotný převod relačních dat do RDF. Z architektonického pohledu tedy reprezentuje databázový a zároveň i převodní modul. Kapitoly SPARQL endpoint a Zpracování RDF výstupu reprezentují Publikační modul architektury. Konečně, kapitola Konfigurace odpovídá konfiguračnímu modulu.

7.2 Jednotné úložiště

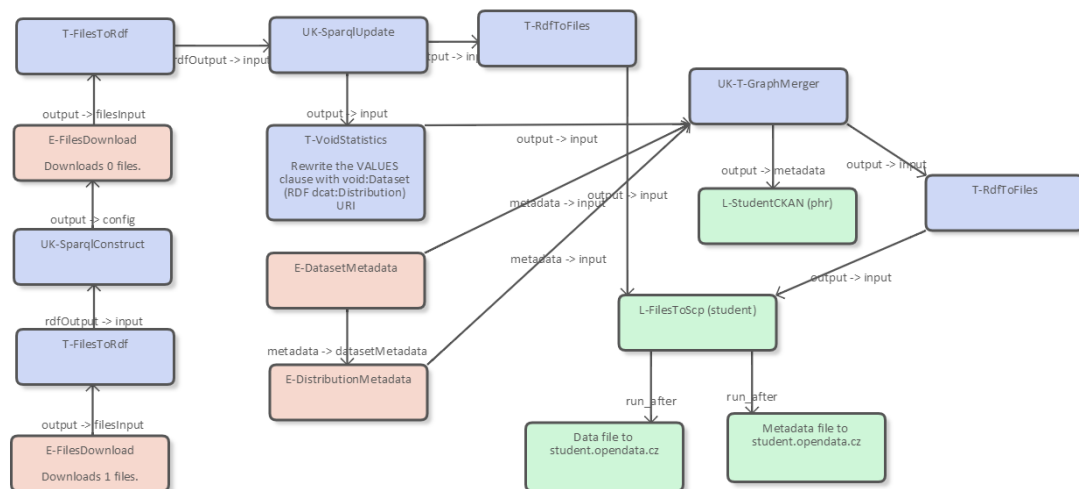
K sběru a zpracování dat využijeme nástroje Unified views. Jedná se o nástroj na jehož vývoji spolupracuje katedra softwarového inženýrství na MFF UK v rámci evropského projektu LOD2[62].

7.2.1 Nástroj Unified views

Nástroj Unified views funguje na bázi zřetěženého zpracování (Pipelining) spojených funkčních jednotek (DPU - Data processing unit). Naším úkolem je vytvořit takovou pipeline, aby se postupně provedly následující kroky:

- Načtení a zpracování definovaného datového katalogu s daty subjektů
- Stažení jednotlivých datasetů a případně provedení operací související s možnou heterogenitou dat
- Uložení předzpracovaných dat do triplestore databáze
- Zpřístupnění dat skrze SPARQL endpoint a registrování datové sady v datovém katalogu

Výsledná pipeline je vidět na Obr. 7.10. Proces zpracování pipeline probíhá po směru šipek.



Obrázek 7.10: Pipeline nad jednotným úložištěm pro zpracování dat o smlouvách

Načtení datového katalogu

K načtení, resp. zpracování datového katalogu s daty subjektů slouží první tři DPU pipeline (viz Obr. 7.10 vlevo dole). V rámci prvního DPU - E-FilesDownload načteme požadovaný datový katalog 6.1. V druhém DPU - T-FilesToRdf převedeme načtená data do RDF reprezentace. Pokud však chceme pomocí nástroje Unified views dávkově stahovat data, je nutné, aby splňovaly strukturu zobrazenou příkladem kódu 7.2. Vytvoříme tedy SPARQL příkaz mapující informace z původního datového katalogu do požadované reprezentace (viz Kód 7.3). Pro tuto funkcionalitu využijeme DPU - UK-SparqlConstruct.

```

1 <http://localhost/resource/config>
2   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://
unifiedviews.eu/ontology/dpu/filesDownload/Config>;
3   <http://unifiedviews.eu/ontology/dpu/filesDownload/hasFile> <
http://localhost/resource/file/0>.
4
5 <http://localhost/resource/file/0>
6   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://
unifiedviews.eu/ontology/dpu/filesDownload/File>;
7   <http://unifiedviews.eu/ontology/dpu/filesDownload/file/uri> "
http://www.zmluva.gov.sk/data/att/117597_dokument.pdf";
8   <http://unifiedviews.eu/ontology/dpu/filesDownload/file/fileName
> "zmluva.pdf".

```

Výpis kódu 7.2: Příklad formátu dat pro dávkové zpracování nástrojem UV

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 PREFIX dct: <http://purl.org/dc/terms/>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX dpu: <http://unifiedviews.eu/ontology/dpu/filesDownload/>
5 PREFIX dpuFile: <http://unifiedviews.eu/ontology/dpu/filesDownload/
file/>
6
7 CONSTRUCT
8 {
9   <http://localhost/resource/config> rdf:type dpu:Config ;
10                                     dpu:hasFile ?distributionUri
11                                     .
12   ?distributionUri rdf:type dpu:File;
13                   dpuFile:uri ?stringUri .
14 }
15 WHERE
16 {
17   ?dataset dcat:distribution ?distribution .
18   ?distribution dct:identifier ?id ;
19               dcat:accessURL ?uri .
20
21   BIND(CONCAT('http://localhost/resource/file/', ?id) as ?
distributionString)
22   BIND(URI(?distributionString) as ?distributionUri)
23   BIND(STR(?uri) as ?stringUri)
24 }
25

```

Výpis kódu 7.3: Příkaz mapující datový katalog do reprezentace nástroje UV

Zpracování jednotlivých datasetů

Nyní pomocí DPU - E-FilesDownload již můžeme stáhnout požadované datasety a pomocí DPU - T-FilesToRdf je převést do RDF reprezentace. V rámci předzpracování provedeme nad daty tuto operaci (v rámci DPU - UK-SparqlUpdate):

- Obecně subjekt publikující smlouvy nutně nemusí mít podrobné informace o smluvních stranách, ale např. má jen IČ. Za předpokladu, že u smluvních stran je vyplněno jen IČ, tak vytvoříme propojení na odpovídající objekt v Linked Data reprezentaci ekonomického subjektu (Viz Kód7.4).

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2
3 WITH <http://tiny.cc/open-contracting>
4 DELETE {?s cn:party ?o .}
5 INSERT {?s cn:party ?newo .}
6 WHERE {
7   ?s cn:party ?o .
8   FILTER(isLiteral(?o) && regex(str(?o), "^[0-9]{8}$"))
9   BIND(CONCAT("http://linked.opendata.cz/resource/business-entity/CZ", ?o) AS ?newo)
10 }

```

Výpis kódu 7.4: Příkaz mapující IČ na reprezentaci ekonomického subjektu

Uložení a publikace výsledné datové sady

V první fázi definujeme metadata o celé datové sadě reprezentující smlouvy. Popíšeme, k čemu datová sada slouží, jaké má URI, licence apod. K tomu slouží DPU - E-DatasetMetadata a DPU - E-DistributionMetadata. Nad datasetem provedeme také statistické výpočty (počty trojic, entit, tříd atd.) pomocí SPARQL příkazu 7.5 v rámci DPU - T-VoidStatistics.

```

1 PREFIX void: <http://rdfs.org/ns/void#>
2 CONSTRUCT
3 {
4   ?ds a void:Dataset;
5     void:triples ?triples;
6     void:entities ?entities;
7     void:classes ?classes;
8     void:properties ?properties;
9     void:distinctSubjects ?dsubjects;
10    void:distinctObjects ?dobjects
11 .
12 }
13 WHERE
14 {
15   { SELECT (COUNT (*) as ?triples) WHERE {?s ?p ?o} }
16   { SELECT (COUNT (distinct ?s) as ?entities) WHERE {?s a ?t} }
17   { SELECT (COUNT (distinct ?t) as ?classes) WHERE {?s a ?t} }
18   { SELECT (COUNT (distinct ?p) as ?properties) WHERE {?s ?p ?o} }
19   { SELECT (COUNT (distinct ?s) as ?dsubjects) WHERE {?s ?p ?o} }
20   { SELECT (COUNT (distinct ?o) as ?dobjects) WHERE {?s ?p ?o} }
21 VALUES ?ds { <http://student.opendata.cz/resource/dataset/phr/contracts/distribution> }
22 }

```

Výpis kódu 7.5: Statistické výpočty nad Otevřenými smlouvami

V druhé fázi výsledky z těchto DPU slijeme dohromady pomocí DPU - UK-T-GraphMerger. Tímto nám vzniknou úplná metadata o datové sadě otevřených smluv. Tyto informace již můžeme zveřejnit v rejstříku datových sad. V našem případě nad platformou CKAN[63] (pomocí DPU - L-StudentCKAN). Datová sada je dostupná na adrese:

- <http://student.opendata.cz/dataset/phr-contracts> [64]

V poslední, třetí fázi data i metadata serializujeme do výstupních souborů v RDF formátu (obě DPU - T-RdfToFiles) a publikujeme do triplestore databáze Virtuoso Universal Server[65] (Zbylé DPU). Vystavený sparql endpoint je dostupný na adrese

- <http://student.opendata.cz/sparql>

7.2.2 Požadavky na architekturu

Databázovým, převodním i publikačním modulem je v našem případě nástroj Unified views. Konfigurací rozumíme jednak nastavení jednotlivých DPU v rámci pipeline, tak načítaný soubor s katalogem požadovaných datových sad. Posledním požadavkem je možnost nastavení intervalu exekuce definované pipeline. V rámci nástroje Unified views k tomu slouží funkce „Schedule a pipeline“ (viz Obr. 7.11).

Obrázek 7.11: Nastavení intervalu exekuce pipeline nástroji UV

7.3 Webová aplikace

7.3.1 Volba technologií a implementační platformy

Webová aplikace je implementována také v technologii ASP.Net se zvoleným architektonickým vzorem MVC. K práci s RDF daty využijeme také knihovnu dotnetRdf. Layout aplikace je tvořen formou responzivního Bootstrap designu. Aplikace se skládá z pěti pohledů: Úvodní obrazovka, Detail subjektu, Detail smlouvy, Veřejné zakázky subjektu a O aplikaci.

7.3.2 Získávání dat

V rámci aplikace využíváme přístup k datovým sadám z těchto SPARQL endpointů:

- Otevřené smlouvy - <http://student.opendata.cz/sparql>
- Organizace, ARES, Orgány veřejné moci - <http://linked.opendata.cz/sparql>
- RÚIAN - <http://ruian.linked.opendata.cz/sparql>
- DBpedia - <http://dbpedia.org/sparql>, nebo česká verze <http://cs.dbpedia.org/sparql>

Konkrétní data se získávají pomocí SPARQL dotazů popsanych níže v rámci popisu jednotlivých pohledů³.

Úvodní obrazovka

Úvodní obrazovku můžeme rozdělit do pomyslných tří částí.

První částí je hlavička obsahující odkazy na web Iniciativy za otevřenou datovou infrastrukturu[8], datový standard pro otevřené smlouvy[43] a informace o aplikaci.

Druhou částí je zobrazení vydavatelů na mapovém podkladu. Nejdříve získáme informace o subjektech pomocí SPARQL dotazu 7.6 (endpoint Otevřené smlouvy). Posléze pro každý subjekt nalezneme jeho link pro přístup k RÚIANU dotazem 7.7 (endpoint Organizace, ARES, Orgány veřejné moci). Pomocí obdržného linku získáme informace o adresním místu z RÚIANu dotazem 7.8 (endpoint RÚIAN). Na závěr zkusíme získat foto subjektu z DBpedia dotazem 7.9 (endpoint DBpedia). Získané informace zobrazíme uživateli na mapovém podkladu. Každý subjekt je zvýrazněn na svých souřadnicích⁴. Po kliku na subjekt se otevře informační okno s podrobnostmi s možností přejítí na detail subjektu.

Třetí částí je seznam smluv. Smlouvy získáme pomocí SPARQL dotazu 7.10 (endpoint Otevřené smlouvy) (viz Obr. 7.12).

³Položky uvedené znakem „@“ jsou proměnné

⁴Počet otevřených smluv je v mapě znázorněn červeným kruhem. Ti, co jich mají více, jsou výraznější.

Otevřené smlouvy

Open data Datový standard O Aplikaci

Seznam smluv

	Vydavatel	Lok. id	Verze	Předmět	Typ smlouvy	Datum vystavení	Datum účinnosti	Částka vč. DPH	Konec účinnosti
Detail	Třebíč	49	1	Stavební práce	Kupní smlouva	18.06.2015	18.06.2015	500000,00	
Detail	Třebíč	49	2	Stavební práce	Kupní smlouva	18.06.2015	04.10.2015	500000,00	
Detail	Třebíč	50	1	Nákup materiálů pro potřeby úřadu	Kupní smlouva	18.06.2015	28.08.2015	10000,00	
Detail	Třebíč	51	1	Softwarová zakázka	Kupní smlouva	18.06.2015	12.09.2015	100000,00	
Detail	Třebíč	53	1	Úklid města	Kupní smlouva	18.06.2015	18.06.2015	12000,00	
Detail	Třebíč	53	2	Úklid města	Kupní smlouva	18.06.2015	18.06.2015	12000,00	
Detail	Třebíč	54	1	Projektová dokumentace divadla	Kupní smlouva	18.06.2015	02.07.2015	20000,00	
Detail	Třebíč	55	1	Koupě nové tramvaje	Kupní smlouva	18.06.2015	18.06.2015	1500000,00	
Detail	Třebíč	55	2	Koupě nové tramvaje C.22	Kupní smlouva	18.06.2015	30.08.2015	1500000,00	
Detail	Třebíč	57	1	Pronájem kinosálu	Nájemní smlouva	18.06.2015	18.06.2015	10000,00	

Počet nalezených smluv: 10

© 2015 - Pavel Hryzlik

OPENDATA CZ

Obrázek 7.12: Úvodní obrazovka webové aplikace

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4
5 SELECT ?Publisher ?Ic
6       (SAMPLE(?subject) AS ?Subject)
7       (SAMPLE(?aresLink ) AS ?AresLink)
8       (COUNT(?Contract) as ?ContractSum)
9 WHERE
10 {
11   ?Contract a cn:Contract ;
12             dc:publisher ?Publisher .
13
14   ?Publisher gr:legalName ?subject ;
15             dc:identifier ?Ic .
16
17   OPTIONAL
18   {
19     ?Publisher owl:sameAs ?aresLink .
20   }
21 }
22
23 GROUP BY ?Publisher ?Ic

```

Výpis kódu 7.6: Získej informace o subjektech

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT *
5 WHERE
6 {
7     @businessEntity s:address ?address .
8
9     ?address ruianlink:adresni-misto ?ruianLink .
10
11     FILTER(CONTAINS(str(?address), 'ares'))
12 }

```

Výpis kódu 7.7: Získej adresní místo

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3 PREFIX ruian: <http://ruian.linked.opendata.cz/ontology/>
4
5 SELECT ?longitude ?latitude
6 WHERE
7 {
8     @addressPoint ruian:adresniBod ?addressPoint .
9
10    ?addressPoint s:geo ?geoCoordinates.
11
12    ?geoCoordinates s:longitude ?longitude ;
13                    s:latitude ?latitude .
14 }

```

Výpis kódu 7.8: Získej polohu subjektu

```

1 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2 PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
3
4 SELECT DISTINCT ?city ?img
5 WHERE {
6     ?city rdfs:label @publisher@cs;
7         dbpedia-owl:thumbnail ?img .
8 }

```

Výpis kódu 7.9: Získej foto subjektu

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4
5 SELECT ?Uri ?Publisher ?PublisherId ?Title ?ContractType ?DateSigned
6        ?ValidFrom ?Amount
7 WHERE
8 {
9     ?Uri a cn:Contract ;
10         dc:title ?Title ;
11         cn:contractType ?ContractType ;
12         dc:created ?DateSigned ;
13         cn:validFrom ?ValidFrom ;
14         dc:publisher ?PublisherLink ;
15         cn:amount ?PriceSpec .

```

```

16 ?PriceSpec gr:hasCurrencyValue ?Amount .
17
18 ?PublisherLink gr:legalName ?Publisher ;
19         dc:identifier ?PublisherId .
20 }

```


Výpis kódu 7.10: Získej všechny smlouvy

Detail subjektu

Detail subjektu nabízí podrobné informace o vydavateli a seznam jeho smluv. Informace o subjektu získáme na základě jeho IČ dotazem 7.11 (endpoint Otevřené smlouvy). Další informace získáme podobně jako na úvodní obrazovce dotazy 7.7,7.9. Jako informaci navíc zkusíme zjistit informace o otevíracích dobách vydavatele dotazem 7.12 (endpoint Organizace, ARES, Orgány veřejné moci). Posléze nad endpointem Otevřené smlouvy obdržíme seznam smluv subjektu pomocí dotazu 7.13 (viz Obr. 7.13).

Otevřené smlouvy Open data Dátový standard O Aplikaci

Město Třebíč




IČ - 00290629 (veřejněné zakázky subjektu)
 Budova - Masarykovo nám. 116/6 PSČ 67401
 Otevírací doba
Pondělí 08:00 - 17:00
Úterý 08:00 - 14:00
Středa 08:00 - 17:00
Čtvrtek 08:00 - 14:00
Pátek 08:00 - 13:00

Budova - Karlovo nám. 104/55 PSČ 67401
 Otevírací doba
Pondělí 08:00 - 17:00
Úterý 08:00 - 14:00
Středa 08:00 - 17:00
Čtvrtek 08:00 - 14:00
Pátek 08:00 - 13:00

Seznam smluv

	Lok. id	Verze	Předmět	Typ smlouvy	Datum vystavení	Datum účinnosti	Konec účinnosti	Částka vč. DPH
Detail	49	1	Stavební práce	Kupní smlouva	18.06.2015	18.06.2015		500000,00
Detail	49	2	Stavební práce	Kupní smlouva	18.06.2015	04.10.2015		500000,00
Detail	50	1	Nákup materiálů pro potřeby úřadu	Kupní smlouva	18.06.2015	28.08.2015		10000,00
Detail	51	1	Softwarová zakázka	Kupní smlouva	18.06.2015	12.09.2015		100000,00
Detail	53	1	Úklid města	Kupní smlouva	18.06.2015	18.06.2015		12000,00
Detail	53	2	Úklid města	Kupní smlouva	18.06.2015	18.06.2015		12000,00
Detail	54	1	Projektová dokumentace divadla	Kupní smlouva	18.06.2015	02.07.2015		20000,00
Detail	55	1	Koupě nové tramvaje	Kupní smlouva	18.06.2015	18.06.2015		1500000,00
Detail	55	2	Koupě nové tramvaje č.22	Kupní smlouva	18.06.2015	30.08.2015		1500000,00
Detail	57	1	Pronájem kinosálu	Nájemní smlouva	18.06.2015	18.06.2015		10000,00

Počet nalezených smluv: 10

© 2015 - Pavel Hryzlik OPENDATA CZ 

Obrázek 7.13: Obrazovka detailu subjektu

```

1 PREFIX dc: <http://purl.org/dc/terms/>
2 PREFIX gr: <http://purl.org/goodrelations/v1#>
3 PREFIX owl: <http://www.w3.org/2002/07/owl#>
4
5 SELECT ?publisher ?aresLink
6 WHERE
7 {
8     ?publisher dc:identifier @ic .
9

```



```

10 OPTIONAL
11 {
12     ?publisher owl:sameAs ?aresLink .
13 }
14 }

```

Výpis kódu 7.11: Získej vydavatele na základě IČ

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?localPlace ?streetAddress ?postalCode ?dayOfWeek ?open ?
   close
5 WHERE
6 {
7     ?subjekt <http://linked.opendata.cz/ontology/domain/seznam.gov.
   cz/ovm/business-entity> @businessEntity ;
8         gr:hasPOS ?localPlace .
9
10    ?localPlace s:address ?address ;
11                gr:openingHoursSpecification ?openingHours .
12
13    ?address s:streetAddress ?streetAddress ;
14             s:postalCode ?postalCode .
15
16    ?openingHours gr:hasOpeningHoursDayOfWeek ?dayOfWeek ;
17                 gr:opens ?open ;
18                 gr:closes ?close .
19 }

```

Výpis kódu 7.12: Získej otevírací hodiny subjektu

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4
5 SELECT ?Uri ?Title ?ContractType ?DateSigned ?ValidFrom ?Amount
6 WHERE
7 {
8     ?Uri a cn:Contract ;
9         dc:title ?Title ;
10        cn:contractType ?ContractType ;
11        dc:created ?DateSigned ;
12        cn:validFrom ?ValidFrom ;
13        dc:publisher ?PublisherLink ;
14        cn:amount ?PriceSpec .
15
16    ?PriceSpec gr:hasCurrencyValue ?Amount .
17
18    ?PublisherLink dc:identifier @publisherId .
19 }

```

Výpis kódu 7.13: Získej všechny smlouvy daného vydavatele

Detail smlouvy

Jak název napovídá, detail smlouvy poskytuje podrobné informace o smlouvě, smluvních stranách, přílohách, dodatcích, milnicích, informacích o ceně a verzích smlouvy. Údaje získáme pomocí dotazů 7.14,7.15,7.16,7.17,7.18,7.19,7.20 nad endpointem Otevřené smlouvy (viz Obr. 7.14).

Otevřené smlouvy Open data Datový standard Aplikaci

Detail smlouvy

Smlouva <http://opendata.trebic.cz/contract/54/1>

Dokument	http://opendata.trebic.cz/file/1247d17e-6cf2-4745-9271-47579bf4def6/Analytics	Kompetence	Soukromoprávní smlouva
Id	Všechny údaje o webu Přehled akvizic 20150301-20150324.pdf	Evidenční číslo veřejné zakázky	
Verze	1	Datum vystavení	18.06.2015
Publikující	Třebíč	Datum účinnosti smlouvy	02.07.2015
Předmět	Projektová dokumentace divadla	Datum ukončení účinnosti smlouvy	
Typ smlouvy	Kupní smlouva	Měna	CZK
Popis předmětu smlouvy	Veřejná služba	Částka bez DPH	
Anonymizováno	false	Částka vč. DPH	20000,00

Smluvní strany

IC	Lokální id	Název	Stát	Plátce DPH	Ulice	Město	PSC	Adresa
371		Katastrální úřad Třebíč	Česká republika	Ne	Funtičkova 12	Třebíč 1, Třebíč	67401	Funtičkova 12 67401 Třebíč 1, Třebíč
393		Fotografia Zlin, s.r.o.	CZE	Ne	Mendlova 9	Zlín	76001	Mendlova 9 76001 Zlín

Přílohy

Dokument	Lokální id	Předmět	Číslo přílohy
http://opendata.trebic.cz/file/42f43d32-7a0d-49c3-ac9a-37fa7a3eb846/Analytics Všechny údaje o webu Přehled publika 20150301-20150324(1).pdf	1	Projektová dokumentace	
http://opendata.trebic.cz/file/60bf83fc-7300-4268-b698-e634bdc22dda/Analytics Všechny údaje o webu Přehled 20150301-20150324 (1).pdf	1	Přílohy	

Verze smlouvy

Číslo verze	Datum publikace	Identifikátor verze
1	18.06.2015 10:59:07	http://opendata.trebic.cz/contract/54/1/version

© 2015 - Pavel Hryzlik OPENDATA CZ

Obrázek 7.14: Obrazovka detailu smlouvy

```
1 SELECT * WHERE { @contract ?p ?o }
```

Výpis kódu 7.14: Získej smlouvu

```
1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4 PREFIX schema: <http://schema.org/>
5
6 SELECT ?Id ?Uri ?Name ?Country ?PaysVat ?StreetAddress ?Locality ?
   PostalCode
7 WHERE
8 {
9   @contract cn:party ?Uri .
10  ?Uri a gr:BusinessEntity ;
11       gr:legalName ?Name ;
12       schema:addressCountry ?Country ;
13       schema:address ?Address ;
14       gr:valueAddedTaxIncluded ?PaysVat .
15
16  OPTIONAL {?Uri dc:identifier ?Id}
17
18  ?Address a schema:PostalAddress ;
19           schema:streetAddress ?StreetAddress ;
```

```

20         schema:addressLocality ?Locality ;
21         schema:postalCode ?PostalCode .
22     }

```

Výpis kódu 7.15: Získej smluvní strany na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX schema: <http://schema.org/>
4
5 SELECT ?Uri ?Title ?Document
6 WHERE
7 {
8     ?Uri a cn:Attachment ;
9         dc:title ?Title ;
10        schema:url ?Document ;
11        cn:contract @contract .
12 }

```

Výpis kódu 7.16: Získej přílohy na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX schema: <http://schema.org/>
4
5 SELECT ?Uri ?Title ?DateSigned ?Document
6 WHERE
7 {
8     ?Uri a cn:Amendment ;
9         dc:title ?Title ;
10        dc:created ?DateSigned ;
11        schema:url ?Document ;
12        cn:contract @contract .
13 }

```

Výpis kódu 7.17: Získej dodatky na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3
4 SELECT ?Uri ?Title ?DueDate
5 WHERE
6 {
7     @contract cn:implementation ?Implementation .
8
9     ?Implementation a cn:Implementation ;
10        cn:milestone ?Uri .
11
12    ?Uri a cn:Milestone ;
13        dc:title ?Title ;
14        cn:dueDate ?DueDate .
15 }

```

Výpis kódu 7.18: Získej milníky na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX gr: <http://purl.org/goodrelations/v1#>
3 SELECT ?Uri ?Amount ?Currency
4 WHERE

```

```

5 {
6   @contract cn:amount ?Uri .
7
8   ?Uri gr:hasCurrencyValue ?Amount ;
9       gr:hasCurrency ?Currency .
10 }

```

Výpis kódu 7.19: Získej informace o ceně

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3
4 SELECT ?Uri ?Issued ?ContractUri ?VersionOrder
5 WHERE
6 {
7   @contract cn:version ?Uri .
8
9   ?Uri dc:issued ?Issued ;
10      cn:uri ?ContractUri ;
11      cn:versionOrder ?VersionOrder .
12
13 FILTER regex(?Contract, @contract)
14 }

```

Výpis kódu 7.20: Získej verze smlouvy

Veřejné zakázky subjektu

Seznam veřejných zakázek je dostupný z detailu subjektu na základě jeho IČ. Získáme jej dotazem 7.21 (endpoint Organizace, ARES, Orgány veřejné moci) (viz Obr. 7.15).

Evidenční číslo ISVZUS	Identifikátor zakázky	Předmět	Dodavatel IČ	Částka vč. DPH
373238	P13V00000086	Zateplení stomatologické polikliniky, Třebíč - dodatečné stavební práce	28359216	95193,12
366631	P13V00000064	Zateplení ZŠ Kpt. Jaroše, Třebíč - dodatečné stavební práce	28359216	91195
358222	P13V00000044	RPS Třebíč - Borovína, Za Rybníkem, et. D (3. etapa) - dodatečné stavební práce	26177005	862678,4
361148	P13V00000050	Dostavba areálu veřejného krytého bazénu Třebíč - dodatečné stavební práce	26230992	7396105,69
211482	P12V00000009	Dostavba areálu veřejného krytého bazénu Třebíč	26230992	67282032
366578	P13V00000063	Úprava ulice Jelinkova, Třebíč - dodatečné stavební práce	29259312	66821
365892	P13V00000060	Zateplení ZŠ Kpt. Jaroše, Třebíč - dodatečné stavební práce	28359216	573569
205340	P12V00000005	ev. č. 12030001 Interaktivní expozice Vnímání všemi smysly	63479982	5687543
371986	P13V00000068	RPS Třebíč - Borovína, Za Rybníkem, et. D (3. etapa) - dodatečné stavební práce	26177005	507055,52
342847	P13V00000012	Údržba veřejné zeleně na části území města Třebíč - západ a zámecký areál	24141623	4962359,6
351400	P13V00000045	Centrální výběr dodavatele elektrické energie pro rok 2014	24128376	4871491,4
375808	P13V00000088	Regenerace panelového sídliště Třebíč - Borovína, Za Rybníkem, etapa C - dodatečné stavební práce	26177005	483844,81
374500	P13V00000087	RPS Třebíč - Borovína, Za Rybníkem, et. D (3. etapa) - dodatečné stavební práce	26177005	36747,7
354922	P13V00000023	Stavební úpravy ul. Kollarova, Třebíč - Borovína	28359216	3016789
358086	P13V00000029	Úprava ulice Jelinkova, Třebíč	29259312	2979095
352897	P13V00000036	Zateplení ZŠ Bartušková, Třebíč	28359216	27338331
366273	P13V00000059	Rozšíření parkoviště v Třebíči, ul. Vitavinská - dodatečné stavební práce	29259312	271801,89
371628	P13V00000085	Zateplení ZŠ Benešova, Třebíč - výměna oken a dveří - dodatečné stavební práce	28359216	233295
13030003	P13V00000011	EV. č. 13030003 - Vestavba M5 v ZŠ Třebíč, ul. Bartušková 700/20	63497786	2006482
375433	P13V00000089	Dostavba areálu veřejného krytého bazénu Třebíč - dodatečné stavební práce	26230992	1973401,34

Počet nalezených veřejných zakázek: 76

1 2 3 4

Obrázek 7.15: Obrazovka seznamu veřejných zakázek subjektu

```

1 PREFIX dc: <http://purl.org/dc/terms/>
2 PREFIX gr: <http://purl.org/goodrelations/v1#>
3 PREFIX pc: <http://purl.org/procurement/public-contracts#>
4 PREFIX pccz: <http://purl.org/procurement/public-contracts-czech#>
5 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
6
7 SELECT DISTINCT ?Uri ?ContractId ?EvNumber ?Title ?SupplierUri ?Id ?
   Amount
8 WHERE
9 {
10   ?Uri pc:contractingAuthority @businessEntity ;
11       dc:title ?Title .
12
13   OPTIONAL
14   {
15     ?Uri pccz:kodprofil ?ContractId ;
16         pccz:kodusvzis ?EvNumber ;
17     ?Tender pco:offeredPrice ?PriceSpec ;
18         pco:supplier ?SupplierUri .
19     ?SupplierUri gr:legalName ?Supplier .
20
21     BIND(CONCAT(str(?SupplierUri), '/identifier ') as ?IcStr)
22     BIND(URI(?IcStr) as ?IcUri)
23
24     ?IcUri skos:notation ?Id .
25     ?PriceSpec gr:hasCurrencyValue ?Amount ;
26         gr:valueAddedTaxIncluded 1 .
27   }
28 }

```

Výpis kódu 7.21: Získej veřejné zakázky na základě subjektu

O aplikaci

V rámci tohoto pohledu jsou uvedeny základní informace o projektu.

7.3.3 Požadavky na architekturu

Pro implementaci jsme zvolili technologii ASP.Net s architektonickým vzorem MVC. Procesním a endpoint modulem je tedy v tomto případě Controller, který na základě klientských požadavků volá odpovídající SPARQL dotazy získávající data z různých zdrojů (endpointů). Komunikace mezi procesním a prezentačním modelem (Komunikační modul) je řešena interně v rámci technologie ASP.Net. V rámci architektury MVC je prezentačním modulem část View obsahující jednotlivé pohledy popsané výše.

8. Evaluace

V rámci této kapitoly se nejdříve zaměříme na otestování konverzního mechanismu platformy nad daty nastiňující reálnou situaci na úřadech, posléze otestováním webové aplikace.

8.1 Test konverzního modulu

V únoru roku 2015 vydalo Ministerstvo vnitra dopadovou studii na odhad nákladů k zavedení zákona o registru smluv[66]. V reakci na tento odhad nedlouho poté vydalo Cetrum aplikované ekonomie o.s. stínový výpočet korigující výsledky Ministerstva vnitra[67]. Na základě těchto studií můžeme získat hrubou představu o tom, kolik jednotlivé subjekty uzavírají přibližně smluv. Veřejné instituce tak rozdělíme do čtyř kategorií:

- Malé - Nejmenší instituce uzavírající jednotky smluv měsíčně s celkovým úhrnem maximálně několika desítek smluv ročně (v rámci měst a obcí jde o nejvyšší zastoupení).
- Střední - Subjekty generující maximálně desítky smluv měsíčně s jednotkami stovek smluv ročně (v rámci všech subjektů pravděpodobně nevýznamnější zastoupení).
- Středně velké - Instituce, které produkují desítky až stovky smluv měsíčně s jednotkami tisíců smluv ročně.
- Velké - Velké instituce se stovkami až tisíci smluv měsíčně s roční produkcí tisíců až desetitisíců smluv.

Pro simulaci prostředí jednotlivých kategorií vytvoříme pro každou skupinu testovací relační databázi s desítkami, stovkami, tisíci a desetitisíci smluv. Nad každou databázi spustíme konverzní modul a změříme dva pravděpodobně nejčastější požadavky - dump dat, resp. výčet všech smluv a vyhledání jedné konkrétní smlouvy. Dump je základní funkcionalitou k vypublikování otevřených smluv. Potřebujeme ho také v rámci platformy, resp. jednotného úložiště, které dílčí dumpy stahuje. Ukázka vyhledání jedné smlouvy slouží spíše k ukázce, že konverzní modul půjde využít i mimo platformu, např. v rámci webových stránek konkrétní veřejné instituce.

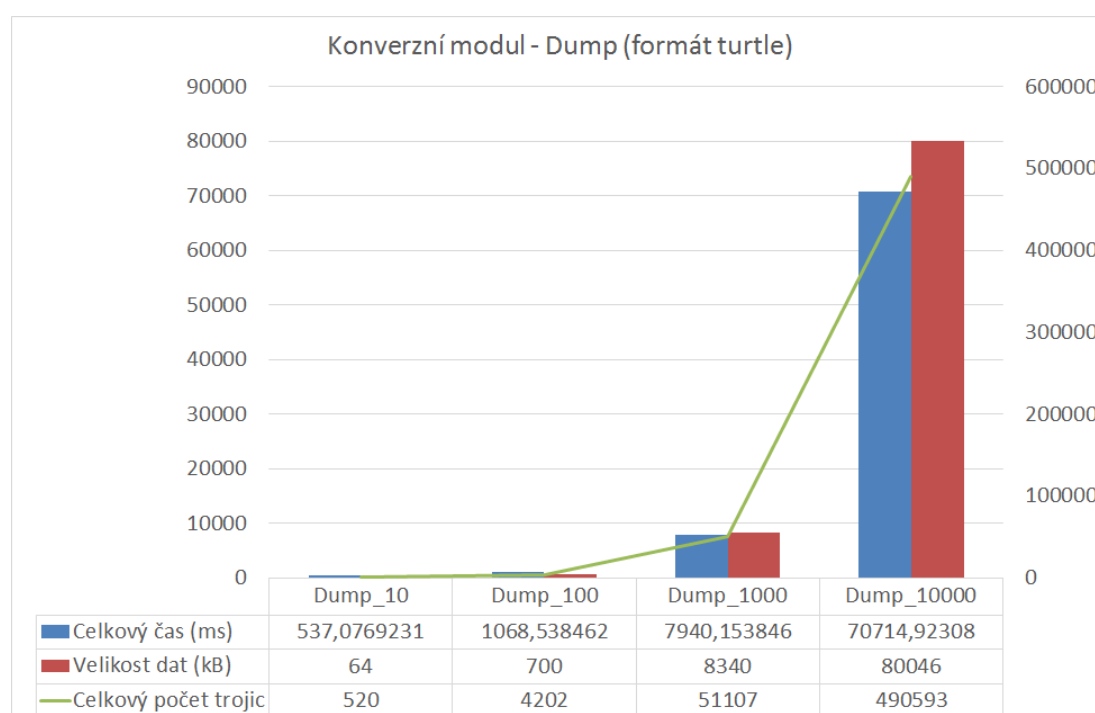
Pro generování dat v SQL databázi byl zvolen nástroj Sql Data Generator. Tento nástroj umožňuje nastavení nejen počtu vygenerovaných dat, ale i např. procentuální zastoupení propojení tabulek nebo šablony pro konkrétní hodnoty v jednotlivých sloupcích. Umožní nám přiblížit se k reálnému obsahu databází veřejných institucí¹. K samotnému profilingu využijeme klasických prostředků prostředí .Net. Změříme dobu od přijetí požadavku po jeho kompletní zpracování.

¹Pro představu je příklad XML scriptu přiložen na datovém nosiči. Sql Data Generator je ale komerční nástroj, který neumožňuje zobrazit generovaný sql příkaz.

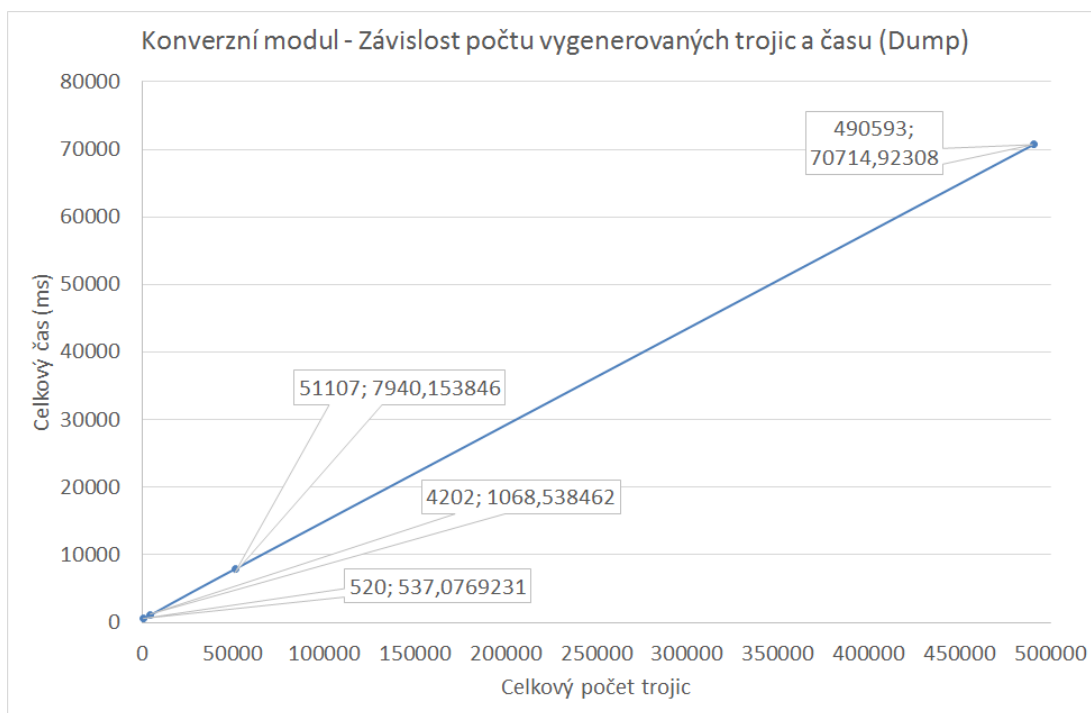
Měření probíhalo na sestavě:

- Intel Core i5-4200U, CPU @ 1.60GHz - 2.30GHz
- 4GB RAM
- 64bit operační systém
- Databáze - MS SQL 2014

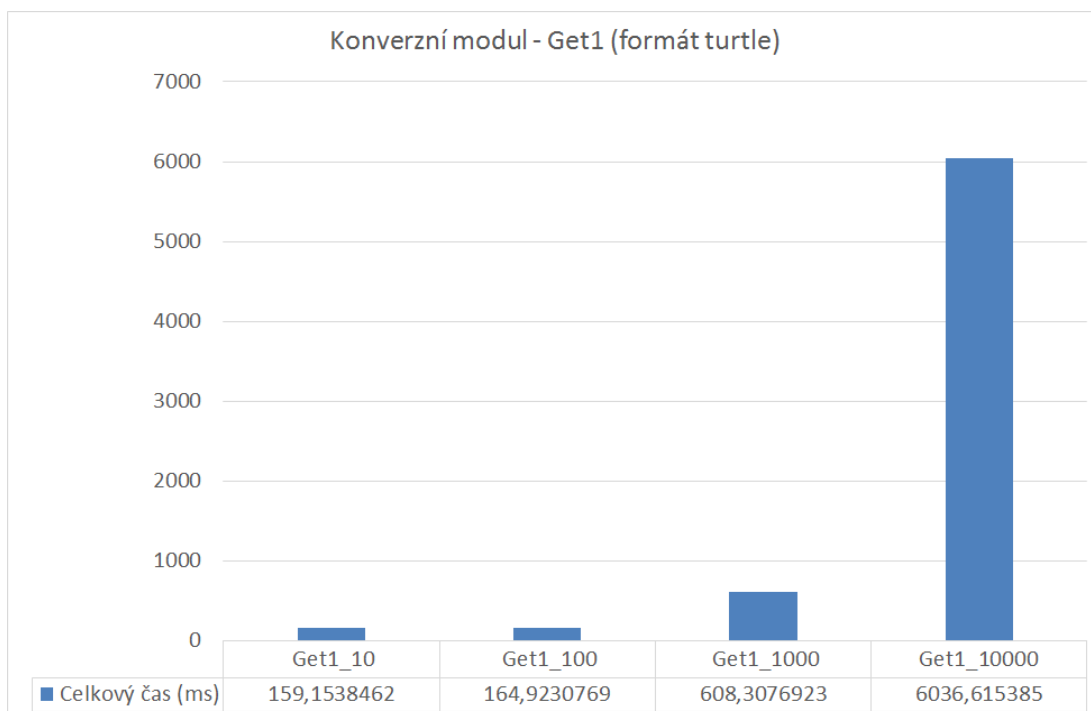
Pro každou kategorii bylo provedeno 15 měření pro dump, resp. vyhledání smlouvy. Z každé sady výsledků se odebrala minimální a maximální hodnota a následně ze zbývajících hodnot byl vypočítán průměr. Pro názornost u dumpu uvádíme také velikost výstupních dat a počet vygenerovaných trojic. Výsledky lze najít v následujících grafech (8.1,8.2,8.3).



Obrázek 8.1: Znázornění časové náročnosti dumpu vybraných dat



Obrázek 8.2: Vztah počtu vygenerovaných trojic a času při dumpu vybraných dat



Obrázek 8.3: Časová náročnost získání jedné smlouvy u vybraných dat

Z výsledků lze konstatovat, že výkon klesá téměř lineárně s množstvím dat (viz graf 8.2). Můžeme říci, že konverzní modul je schopen poskytovat základní funkcionalitu v rozumném čase u menších, středních i středně velkých institucí. U velkých institucí už výsledky nejsou ideální. Institucí publikujících takové množství smluv je ale v prostředí České republiky velmi málo. Příslibem je také to, že využívaný R2RML procesor podléhá soustavnému vývoji a do budoucna lze očekávat výrazné zrychlení.

8.2 Test webové aplikace

Webová aplikace se skládá z pěti pohledů, kde čtyři z nich volají jinou sadu SPARQL dotazů nad různými endpointy (viz kapitola Implementace). Řekněme, že úzkým hrdlem aplikace je rychlost zpracování dotazů nad jednotlivými endpointy, zvláště pak nad endpointem jednotného úložiště platformy. Provedeme proto 4 různé testy simulující dotazy již zmíněných pohledů. Každý test spustíme 50x a podobně jako u konverzního modulu odebereme minimální a maximální hodnotu a ze zbývajících hodnot vypočítáme průměr. V jednotném úložišti je uložen datový soubor s cca 10000 smlouvami (řádově statisíce trojic). Výsledky jednotlivých testů vidíme v následující tabulce 8.1:

Test	Celkový čas(ms)
Test1 - Hlavní obrazovka	4853,6667
Test2 - Detail vydavatele	92,9584
Test3 - Detail smlouvy	49,6875
Test3 - Veřejné zakázky vydavatele	13,1667

Tabulka 8.1: Výsledky testování webové aplikace

Z výsledků je zřejmé, že načítání velkého množství dat v rámci hlavní obrazovky je výrazně pomalejší, než u zbylých pohledů². Nutno podotknout, že aplikace nebyla testována na profesionálním řešení, ale v domácích podmínkách. Přesto lze konstatovat, že načítání tisíců až desetitisíců smluv v rámci jednotek sekund lze považovat za rozumné. Typickým možným zrychlením, které můžeme vidět třeba na portálu veřejné správy[39], je rozlišování seznamu smluv podle roků a měsíců.

²U detailu vydavatele také záleží na počtu jeho smluv.

9. Shrnutí procesu otevírání smluv

Za účelem větší názornosti shrneme dosavadní proces otevírání smluv jedním diagramem (viz Obr. 9.1). Diagram je rozdělen do tří řádků a tří sloupců. První řádek zachycuje proces otevírání dat. Druhý řádek znázorňuje otevírání dat s využitím principů Linked Data. Třetím řádkem je zapojení relačních dat do procesu. V prvním sloupci se nacházíme na úrovni schématu. Zde definujeme standardy, ontologie a schémata. Druhý sloupec znázorňuje produkci otevřených a propojitelných dat. V třetím sloupci jsme na úrovni publikace dat, resp. serializace otevřených a propojitelných dat do přenositelných formátů.

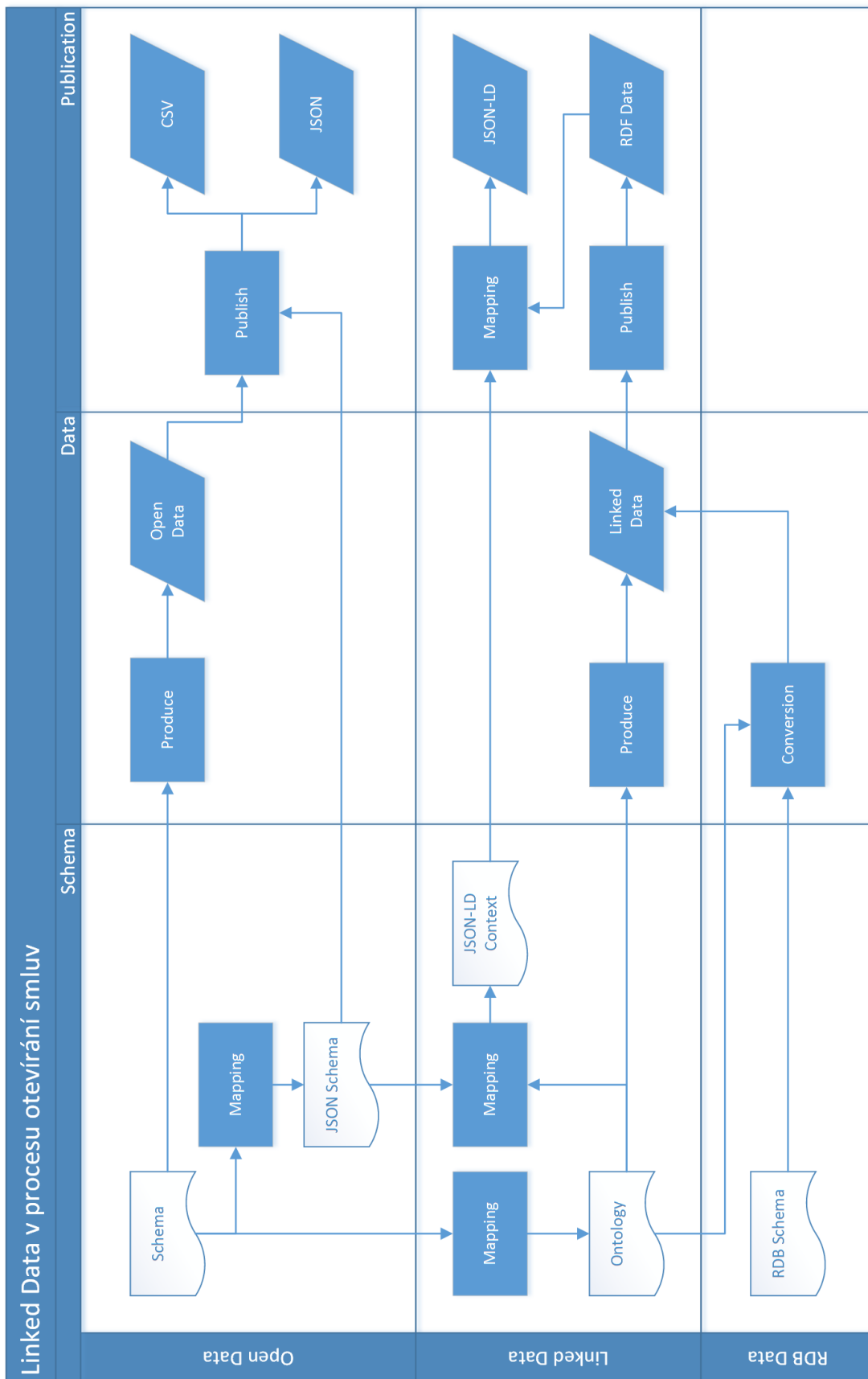
Začátek procesu je v tvorbě schématu. V našem případě se jedná o datový standard pro otevřené smlouvy. Na základě schématu je v diagramu znázorněna možnost produkovat otevřená data. Ta jsou pak serializovatelná do konkrétních datových formátů. Lepšími datovými formáty jsou ale ty, které jsou definované pomocí schématu datového, pro naše účely se jedná o formát JSON. Z diagramu je vidět, že ze schématu vytvoříme datové JSON schéma, které je poté využitelné při publikaci dat.

Na základě schématu můžeme také nadefinovat RDF ontologii, díky které se dostaneme do světa Linked Data. V diagramu je obdobně jako u otevřených dat znázorněna možnost na základě ontologií produkovat Linked Data a ta pak serializovat do RDF datových formátů.

Speciálním případem serializace RDF dat jsou data ve formátu JSON-LD. Na základě schématu, resp. datového JSON schématu a RDF ontologie, můžeme vytvořit takový JSON-LD kontext, že výsledná vypublikovaná JSON-LD data budou naplňovat jak datový standard, tak i RDF ontologii.

Posledním krokem je zapojení relačních dat do procesu otevírání smluv. Tato relační data na základě relačního schématu, resp. jeho datového modelu a RDF ontologie, zkonvertujeme do Linked Data.

Lze si všimnout, že se ne bavíme pouze o údajích o smlouvách. Tento postup lze obecně použít nejen pro smlouvy, ale i pro libovolnou jinou doménu.



Obrázek 9.1: Linked Data v procesu otevírání smluv

Závěr

V rámci této práce jsme si kladli za cíl využít principů Linked Data pro publikaci a sdílení dat o smlouvách.

Začali jsme definováním datového standardu pro otevřené smlouvy. Ten probíhal v rámci akční skupiny pod záštitou Oživení o.s. a Centra aplikované ekonomie o.s. Hlavním přínosem je reálná možnost zařazení standardu do sady doporučení Ministerstva vnitra pro publikovatelné datové sady. Na základě standardu byla navržena podoba datových formátů pro jejich publikaci. Dílčím výsledkem byla tvorba metodiky ve formě webové aplikace mající za cíl technicky i věcně datový standard popsat.

V dalším kroku byla navržena ontologie pro publikaci otevřených smluv v RDF podobě. Zaměřili jsme se také na možnost propojení se souvisejícími daty. Ukázali jsme výhody serializace RDF dat v JSON-LD formátu. Klíčovým přínosem JSON-LD formátu je, že vypublikovaná data splňují datový standard pro otevřené smlouvy a zároveň se jedná o RDF data.

V následující části jsme navrhli a implementovali platformu pro otevírání smluv. Platforma je složena ze třech základních součástí: Konverzního modulu, jednotného úložiště a prezentační webové aplikace.

V návrhu konverzního modulu jsme se zaměřili na konverzi relačních dat stávajících informačních systémů do RDF podoby splňující principy Linked Data. Jako zdroj pro konkrétní implementaci byl zvolen modul Munis ESML informačního systému Triada spol. s.r.o. Řešení přináší zajímavý přístup mapování relačních dat do RDF podoby pomocí R2RML skriptu. Díky tomu lze konverzní mechanismus s drobnými úpravami využít i nad jinými informačními systémy. Druhou součástí platformy bylo navrženo a implementováno jednotné úložiště. Úložiště je na základě definovaného datového katalogu schopno stahovat konkrétní datasety údajů o smlouvách v RDF podobě a ukládat je do triplestore databáze. Díky navržené jednoznačné identifikaci entit odpadly problémy s heterogenitou dat.

Jako poslední součást platformy byla navržena a implementována webová aplikace zpřístupňující údaje o smlouvách z jednotného úložiště koncovým uživatelům. V rámci aplikace jsme se zaměřili na demonstraci přínosů využití principů Linked Data. Navrhli jsme proto síť propojených datasetů s cílem poskytnout uživateli údaje o smlouvách obohacených o informace např. z ARESu, RUIANU, nebo Věstníku veřejných zakázek.

Následně jsme otestovali konverzní mechanismus a webovou aplikaci ve snaze simulovat možnosti reálného využití. Na základě procesu otevírání smluv jsme také uvedli obecný postup otevírání dat využitelný i v jiných doménách.

Linked Data v procesu otevírání smluv

V rámci této práce jsme ukázali, že využití principů Linked Data je pro doménu smluvních údajů možné. Ukázali jsme také postup, jak toho dosáhnout. Shrňme si tedy základní výhody a nevýhody využití Linked Data v procesu otevírání smluv:

Výhody

- V našem případě se zároveň jedná o otevřená data. Údaje o smlouvách tedy mohou být dostupné široké veřejnosti na internetu a přinášet veškeré výhody otevřených dat.
- Díky možnosti propojení se smlouvy stanou součástí daleko širšího kontextu otevřených a propojitelných dat. Zvýší se tak informační hodnota každé smlouvy
- Údaje o každé smlouvě, resp. entitě jsou dostupné pod vlastním URI. Smlouva je tak na jednom místě a můžeme se na ni libovolně odkazovat.

Nevýhody

- Nelze očekávat, že práce nad daty využívajícími principy Linked Data, bude rychlá jako nad relačními databázemi.
- Častou nevýhodou využití principu Linked Data bývá velká náročnost kladená na subjekt, který chce zveřejňovat (v rámci platformy navržený konverzní mechanismus ale nároky na subjekt výrazně redukuje).
- Obecně příprava dat, tvorba standardu, ontologie, definování URI identifikátorů apod. vyžaduje jisté znalosti a netriviální úsilí.

K přípravě dat bych rád doplnil, že před zpracováním podobných domén jako jsou údaje o smlouvách do Linked Data podoby, je důležité navrhnout datový standard definující, co je vůbec vybrané domény obsahem. Ze zkušenosti v rámci akční skupiny pro tvorbu standardu mohu konstatovat, že tato činnost nemusí být triviální. Každá konkrétní položka může mít různé technické, ale především právní aspekty, které je třeba podrobit diskusi s relevantními osobami.

S ohledem na potřebnou přípravu dat se tedy nabízí otázka celkové pracnosti. Náročnost přípravy dat a implementace konverzního modulu bych na základě zkušenosti odhadl zhruba takto:

Datový standard	Linked Data	Konverzní modul + R2ML mapování
2 člověkoměsíce	1,5 člověkoměsíce	2,5 člověkoměsíce
33,3%	25%	41,7%

Celkovou náročnost otevření této domény smluv tedy můžeme odhadnout na zhruba 6 člověkoměsíců. Pro každý další subjekt zapracovávající doménu smluv pak stačí odhadem 2,5 člověkoměsíce (tvorba konverzního modulu a R2RML mapování).

Seznam zdrojů a použité literatury

- [1] *Předpis č. 106/1999 Sb. Zákon o svobodném přístupu k informacím* [online]. [cit. 2015-11-30] Dostupné na: <http://www.zakonyprolidi.cz/cs/1999-106>
- [2] *Směrnice Evropského parlamentu a Rady 2013/37/EU ze dne 26. června 2013, kterou se mění směrnice 2003/98/ES o opakovaném použití informací veřejného sektoru Text s významem pro EHP* [online]. [cit. 2015-11-30] Dostupné na: <http://www.eurlex.cz/dokument.aspx?celex=32013L0037>
- [3] *Ministerstvo vnitra - Otevřená data* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.mvcr.cz/clanek/otevrena-data.aspx?q=Y2hudW09Mg%3d%3d>
- [4] *Rekonstrukce státu* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.rekonstrukcestatu.cz/>
- [5] *Fond Otakara Motejla* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.motejl.cz/>
- [6] *Oživení o.s.* [online]. 2013, [cit. 2015-11-30] Dostupné na: <http://www.oziveni.cz/>
- [7] *Fórum pro otevřená data* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.otevrenadata.cz/o-nas/forum-pro-otevrena-data/>
- [8] *Iniciativa za otevřenou datovou infrastrukturu* [online]. [cit. 2015-11-30] Dostupné na: <http://opendata.cz/>
- [9] *Návrh zákona o registru smluv a o změně zákona č. 137/2006 Sb., o veřejných zakázkách, ve znění pozdějších předpisů - tisk 42* [online]. [cit. 2015-11-30] Dostupné na: <http://www.psp.cz/sqw/historie.sqw?o=7&T=42>
- [10] CHLAPEK, D., KUČERA, J., NEČASKÝ, M., KUBÁŇ, M. *Open data and PSI in the Czech Republic* [online]. 2014, [cit. 2015-11-30] Dostupné na: <http://www.epsiplatform.eu/content/open-data-and-psi-czech-republic>
- [11] BERG, M., BOČEK, J., BOUCHAL, P., MRÁČEK, J., NEČASKÝ, M. *Otevřená data ve státní správě: Nová éra rozhodování* [online]. 2012, ISBN: 978-80-87110-24-9, [cit. 2015-11-30] Dostupné na: <http://www.osf.cz/publikace/otevrena-data-ve-statni-sprave-nova-era-rozhodovani/>
- [12] BOČEK, J., MRÁČEK, J., Mynarz, J. *Otevřená data: Příležitost pro Českou republiku* [online]. 2012, ISBN: 978-80-87725-03-0, [cit. 2015-11-30] Dostupné na: <http://www.osf.cz/publikace/otevrena-data-prilezitost-pro-ceskou-republiku/>
- [13] *Školení otevřených dat VS ČR* [online]. 2015, [cit. 2015-11-30] Dostupné na: http://opendata.gov.cz/_media/edu:skoleni_open_data_final.pdf

- [14] *Starostové pro transparentnost* [online]. 2014, [cit. 2015-11-30] Dostupné na: <http://starostoveprotransparentnost.cz/>
- [15] *EconLab (dříve Centrum aplikované ekonomie)* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.econlab.cz/>
- [16] BOČEK, J., ČEPICKÝ, J., MRÁČEK, J. *Jak otevírat data?* [online]. 2014, ISBN 978-80-87725-15-3, [cit. 2015-11-30] Dostupné na: <http://www.otevrenadata.cz/res/data/001/003498.pdf>
- [17] BERNERS-LEE, T. *5★ Open Data* [online], 2006. [cit. 2015-11-30] Dostupné z <http://5stardata.info/en/>
- [18] BERNERS-LEE, T. *Linked Data - Design Issues* [online], 2006. [cit. 2015-11-30] Dostupné z <http://www.w3.org/DesignIssues/LinkedData.html>
- [19] *Office Open XML* [online]. Ecma International, 1999, [cit. 2015-11-30] Dostupné na: <http://www.ecma-international.org/publications/standards/Ecma-376.htm>
- [20] *Semantic web* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/standards/semanticweb/>
- [21] *RDF 1.1 Concepts and Abstract Syntax* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>
- [22] *SPARQL 1.1 Query Language* [online]. W3C Recommendation, 2013, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>
- [23] *The Linking Open Data cloud diagram* [online]. 2014, [cit. 2015-11-30] Dostupné na: <http://lod-cloud.net/>
- [24] *Case study on how Linked Data is transforming eGovernment* [online]. 2013, [cit. 2015-11-30] Dostupné na: <https://joinup.ec.europa.eu/community/semic/document/case-study-how-linked-data-transforming-egovernment>
- [25] KUČERA, J., CHLAPEK, D. *Benefits and Risks of Open Government Data* [online]. 2014, [cit. 2015-11-30] Dostupné na: <http://www.si-journal.org/index.php/JSI/article/viewFile/185/136>
- [26] *Dublin core ontology* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://purl.org/dc/terms/>
- [27] *Friend-of-a-Friend ontology* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://xmlns.com/foaf/0.1/>
- [28] *Schema ontology* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://schema.org/>
- [29] *Linked Open Vocabularies* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://lov.okfn.org/dataset/lov/>

- [30] *OWL 2 Web Ontology Language Document Overview (Second Edition)* [online]. W3C Recommendation, 2012, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
- [31] *RDF Schema* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/rdf-schema/>
- [32] *RDF 1.1 N-Triples* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/n-triples/>
- [33] *RDF 1.1 N-Quads* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/n-quads/>
- [34] *RDF/XML Syntax Specification* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/REC-rdf-syntax>
- [35] *RDF 1.1 Turtle* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/turtle/>
- [36] *RDF 1.1 TriG* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/trig>
- [37] *RDFa Core 1.1* [online]. W3C Recommendation, 2015, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/rdfa-syntax/>
- [38] *JSON-LD 1.0* [online]. W3C Recommendation, 2014, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/json-ld/>
- [39] *Portál veřejné správy* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://portal.gov.cz>
- [40] *Standardy publikace a katalogizace otevřených dat veřejné správy ČR* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://opendata.gov.cz/>
- [41] *Původní koncept datového standardu pro otevřené smlouvy* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.bezkorupce.cz/wp-content/uploads/2014/08/Datov%C3%BD-standard- pro-registr-smluv1.pdf>
- [42] *Otevřená města* [online]. 2014, [cit. 2015-11-30] Dostupné na: <http://www.otevrenamesta.cz/>
- [43] *Metodika zveřejňování smluv* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://standard.zindex.cz/>
- [44] *Portál informačního systému o veřejných zakázkách - Číselníky a klasifikace* [online]. 2015, [cit. 2015-11-30] Dostupné na: http://www.isvz.cz/ISVZ/Ciselniky/ISVZ_klasifikace_ciselniky.aspx
- [45] *JSON* [online]. Ecma International, 1999, [cit. 2015-11-30] Dostupné na: <http://json.org/>
- [46] *CSV* [online]. 2005, [cit. 2015-11-30] Dostupné na: <https://tools.ietf.org/html/rfc4180>

- [47] CHLAPEK, D., KUČERA, J., NEČASKÝ, M. *Metodika publikace otevřených dat veřejné správy ČR* [online]. 2012, [cit. 2015-11-30] Dostupné na: http://www.korupce.cz/assets/partnerstvi-pro-otevrenevladnuti/otevrena-data/Metodika_Publ_OpenData_verze_1_0.pdf
- [48] *JSONSchema* [online]. Internet Engineering Task Force , 2013, [cit. 2015-11-30] Dostupné na: <http://json-schema.org/documentation.html>
- [49] *Otevřené smlouvy - JSON Schema* [online]. 2015, [cit. 2015-11-30] Dostupné na: https://raw.githubusercontent.com/PavelHryzlik/ContractStandard/master/standard/schema/contract_schema.json
- [50] *Dokuwiki - Open Source wiki software* [online]. 2015, [cit. 2015-11-30] Dostupné na: <https://www.dokuwiki.org/>
- [51] *The Open Contracting Data Standard* [online]. [cit. 2015-11-30] Dostupné na: <http://standard.open-contracting.org/>
- [52] *Commerce Vocabulary* [online]. 2015, [cit. 2015-11-30] Dostupné na: <https://web-payments.org/vocabs/commerce>
- [53] *GoodRelations Vocabulary* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.heppnetz.de/ontologies/goodrelations/>
- [54] *VANN: A vocabulary for annotating vocabulary descriptions* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://vocab.org/vann/>
- [55] *R2RML: RDB to RDF Mapping Language* [online]. W3C Recommendation, 2012, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/r2rml/>
- [56] *D2RQ* [online]. 2012, [cit. 2015-11-30] Dostupné na: <http://d2rq.org/d2rq-language>
- [57] *Předpis č. 101/2000 Sb. Zákon o ochraně osobních údajů a o změně některých zákonů* [online]. [cit. 2015-11-30] Dostupné na: <http://www.zakonyprolidi.cz/cs/2000-101>
- [58] *Projekt DotNetR2RMLStore* [online]. 2014, [cit. 2015-11-30] Dostupné na: <https://github.com/mchaloupka/DotNetR2RMLStore>
- [59] CHALOUPKA, M. *Querying RDF graphs stored in a relational database using SPARQL and R2RML* [online]. 2014, [cit. 2015-11-30] Dostupné na: <https://is.cuni.cz/webapps/zzp/detail/143369/>
- [60] *Date and Time Formats* [online]. W3C Note, 1997, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/NOTE-datetime>
- [61] *SPARQL result formats* [online]. W3C Recommendation, 2013, [cit. 2015-11-30] Dostupné na: <http://www.w3.org/TR/sparql11-overview/#sparql11-results>
- [62] *LOD2 - Creating Knowledge out of Interlinked Data* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://lod2.eu/Welcome.html>

- [63] *CKAN, open-source data portal platform* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://ckan.org/>
- [64] *Otevřené smlouvy - Datový katalog CKAN* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://student.opendata.cz/dataset/phr-contracts>
- [65] *OpenLink - Virtuoso* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://virtuoso.openlinksw.com/>
- [66] *DOPADOVÁ STUDIE ke komplexnímu pozměňovacímu návrhu k návrhu poslanců Jana Farského, Andreje Babiše, Pavla Bělobrádka a dalších na vydání zákona o Registru smluv a o změně zákona č. 137/2006 Sb., o veřejných zakázkách, ve znění pozdějších předpisů (sněmovní tisk 42, VII. volební období Poslanecké sněmovny Parlamentu České republiky)* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.janfarsky.cz/wp-content/uploads/2015/05/Dopadov%C3%A1-studie-ke-KPN-k-registru-smluv-PRACOVNI-VERZE-27-02-2015-1.pdf>
- [67] *Stínový výpočet RIA k návrhu zákona o registru smluv* [online]. 2015, [cit. 2015-11-30] Dostupné na: <http://www.rekonstrukcestatu.cz/publikace/2015-03-04-stinovy-vypocet-ria-k-registru-smluv.pdf>

Seznam obrázků

2.1	Logo otevřených dat	7
2.2	Stupně otevřenosti dat	9
2.3	Linked Open Data Cloud	11
2.4	Základní RDF trojice	12
2.5	Jednoduchý RDF graf	13
2.6	RDF graf s přiřazenými typy	14
2.7	Ontologie třídy <i>Contract</i>	15
2.8	Odpovídající si entity	16
3.1	Datový standard pro zveřejňování smluv - UML diagram	22
3.2	Metodika zveřejňování smluv	39
6.1	Základní pohled na platformu otevřených smluv (Logical view)	61
6.2	Rozdělení platformy do modulů (Decomposition view)	62
6.3	Datová síť	66
6.4	Obohacený kontext smluv díky propojeným datům	67
7.1	Modul ESML	68
7.2	Zjednodušený datový model (bez atributů) Munis ESML	70
7.3	R2RML mapování - tabulky k mapování smlouvy	71
7.4	R2RML mapování externího kontaktu	74
7.5	R2RML mapování vlastností Smluvní strany a Adresy	76
7.6	R2RML mapování vlastností Příloha	77
7.7	R2RML mapování vlastností Verze	79
7.8	R2RML mapování vlastností Milníku	80
7.9	R2RML mapování vlastností Vydavatele	81
7.10	Pipeline nad jednotným úložištěm pro zpracování dat o smlouvách	85
7.11	Nastavení intervalu exekuce pipeline nástroji UV	88
7.12	Úvodní obrazovka webové aplikace	90
7.13	Obrazovka detailu subjektu	92
7.14	Obrazovka detailu smlouvy	94
7.15	Obrazovka seznamu veřejných zakázek subjektu	96
8.1	Znázornění časové náročnosti dumpu vybraných dat	99
8.2	Vztah počtu vygenerovaných trojic a času při dumpu vybraných dat	100
8.3	Časová náročnost získání jedné smlouvy u vybraných dat	100
9.1	Linked Data v procesu otevírání smluv	103

Seznam tabulek

3.1	Položky tabulek datového standardu	23
3.2	Validita	24
3.3	Akceptovatelné soubory	24
3.4	Vlastnosti dokumentu	25
3.5	Vlastnosti vydavatele	25
3.6	Vlastnosti verze smlouvy	26
3.7	Vlastnosti smlouvy	27
3.8	Vlastnosti přílohy	27
3.9	Vlastnosti dodatku	27
3.10	Vlastnosti smluvní strany	28
3.11	Vlastnosti nadřazené instituce	28
3.12	Vlastnosti adresy	29
3.13	Vlastnosti objednávky	29
3.14	Vlastnosti faktury	29
3.15	Vlastnosti implementace	30
3.16	Vlastnosti milníku	30
3.17	Vlastnosti transakce	30
3.18	Číselník typu dokumentu	31
3.19	Číselník typu smlouvy	31
3.20	Číselník datového souboru	33
3.21	Datový standard serializovaný do CSV	38
4.1	Mapování entity Document	45
4.2	Mapování entity Vydavatel	45
4.3	Mapování entity Verze smlouvy	46
4.4	Mapování entity Smlouva	46
4.5	Mapování entity Příloha	47
4.6	Mapování entity Dodatek	47
4.7	Mapování entity Smluvní strana	47
4.8	Mapování entity Nadřazená instituce	48
4.9	Mapování entity Nadřazená instituce	48
4.10	Mapování entity Objednávka	48
4.11	Mapování entity Faktura	49
4.12	Mapování entity Implementace	49
4.13	Mapování entity Milník	49
4.14	Mapování entity Transakce	50
8.1	Výsledky testování webové aplikace	101

Výpisy kódu

2.1	Příklad RDF dat - N-Triples	16
2.2	Příklad RDF dat - Turtle	17
2.3	Příklad RDF Ontologie - Turtle	18
2.4	Obyčejný JSON dokument	19
2.5	Příklad RDF dat - JSON-LD	19
2.6	Příklad RDF dat - JSON-LD s Contextem	19
3.1	JSON soubor s jednou smlouvou	33
4.1	JSON-LD Context	52
4.2	JSON-LD Soubor s jednou smlouvou	54
6.1	Datový katalog pro jednotné úložiště	64
7.1	Rozšíření knihovny JSON-LD.Net	84
7.2	Příklad formátu dat pro dávkové zpracování nástrojem UV	86
7.3	Příkaz mapující datový katalog do reprezentace nástroje UV	86
7.4	Příkaz mapující IČ na reprezentaci ekonomického subjektu	87
7.5	Statistické výpočty nad Otevřenými smlouvami	87
7.6	Získej informace o subjektech	90
7.7	Získej adresní místo	90
7.8	Získej polohu subjektu	91
7.9	Získej foto subjektu	91
7.10	Získej všechny smlouvy	91
7.11	Získej vydavatele na základě IČ	92
7.12	Získej otevírací hodiny subjektu	93
7.13	Získej všechny smlouvy daného vydavatele	93
7.14	Získej smlouvu	94
7.15	Získej smluvní strany na základě smlouvy	94
7.16	Získej přílohy na základě smlouvy	95
7.17	Získej dodatky na základě smlouvy	95
7.18	Získej milníky na základě smlouvy	95
7.19	Získej informace o ceně	95
7.20	Získej verze smlouvy	96
7.21	Získej veřejné zakázky na základě subjektu	97

Přílohy

A Příloha

Harmonogram událostí v souvislosti s otevřenými smlouvami

- 2.12.2013** Předložen návrh zákona o Registru smluv
- 26.11.2014** Seminář Transparentnost v obcích - Myšlenka datového standardu pro otevřené smlouvy
- 4.12.2014** Schůzka akční skupiny k tvorbě datového standardu pro otevřené smlouvy na radnici Praha 6
- 6.1.2015** Schůzka akční skupiny k potvrzení datového standardu pro otevřené smlouvy na radnici Praha 6
- 6.2.2015** Představen spolek Otevřená města
- 18.3.2015** Schůzka s Jiřím Skuhrovcem k projektu Metodika zveřejňování smluv
- 29.8.2015** Odeslán datový standard pro otevřené smlouvy Ministerstvu vnitra (ve formě CSV)
- 18.9.2015** Schválen zákon o Registru smluv poslaneckou sněmovnou
- 21.10.2015** Zakládající konference spolku Otevřená města
- 22.10.2015** Zákon vrácen senátem s pozměňovacími návrhy
- 24.11.2015** Sněmovna setrvala na původním návrhu zákona

B Příloha

Uživatelská dokumentace

Adresa konverzního mechanismu je nastavena na:

- *http://[domain]/sparql*

Základní funkcionalita konverzního modulu a webové aplikace je k nalezení v kapitole Implementace platformy. U obou projektů stačí k běžnému nastavení soubor *Web.config*. Projekty byly vyvíjeny ve vývojovém prostředí Visual Studio 2013, později ve verzi 2015. Základní předpoklady pro využití platformy jsou:

- Konverzní mechanismus
 - Prostředí, kde lze publikovat webovou aplikaci v prostředí .NET, tedy Windows, Windows Server, IIS, MS Azure apod.
 - MSSQL Server 2012 a vyšší pro funkci R2RML procesoru
 - Přístup k MSSQL databázi firmy Triada, spol. s r. o.
 - Přístup k datovému úložišti firmy Triada, spol. s r. o.
 - Knihovna pro práci s datovým úložištěm
 - R2RML mapovací skript
- Jednotné úložiště
 - Nástroj UnifiedViews
 - Triplestore databáze, např. Openlink Virtuoso Universal Server
 - Konfigurační soubor s datovým katalogem požadovaných datasetů
- Webová aplikace
 - Prostředí, kde lze publikovat webovou aplikaci v prostředí .NET, tedy Windows, Windows Server, IIS, MS Azure apod.
 - Přístup k SPARQL endpointům:
 - * *http://student.opendata.cz/sparql*
 - * *http://linked.opendata.cz/sparql*
 - * *http://ruian.linked.opendata.cz/sparql*
 - * *http://cs.dbpedia.org/sparql*

C Příloha

Obsah přiloženého datového nosiče

Základní struktura přiloženého datového nosiče je rozdělena do těchto složek (zmíníme také klíčové skripty):

<code>/ContractStandard/</code>	Složka datového standardu
<code>/ContractStandard/lod/</code>	Složka se skripty využívající principy LinkedData
<code>/ContractStandard/lod/contract_context.jsonld</code>	JSON-LD Context - mapující skript
<code>/ContractStandard/lod/contract_ontology.ttl</code>	Ontologie pro otevřené smlouvy
<code>/ContractStandard/lod/subject_catalog.ttl</code>	Datový katalog pro UnifiedViews
<code>/ContractStandard/lod/triada_esmluv_r2rml.ttl</code>	R2RML mapovací skript
<code>/ContractStandard/samples/</code>	Složka s příklady otevřených smluv
<code>/ContractStandard/schema/</code>	Složka se schémata datového standardu
<code>/ContractStandard/schema/contract_schema.csv</code>	Datový standard pro otevřené smlouvy - CSV
<code>/ContractStandard/schema/contract_schema.json</code>	Datový standard pro otevřené smlouvy - JSON Schema
<code>/ContractViewer/</code>	Složka s projektem webové aplikace
<code>/TestResults/</code>	Složka s testy konverzního modulu a web. aplikace
<code>/TriadaEndpoint/</code>	Složka s projektem konverzního modulu
<code>/UnifiedViews/</code>	Složka se skripty použité v rámci pipeline
<code>/UnifiedViews/UnifiedViewsExport/</code>	Vyexportované soubory popisující pipeline nástroje UnifiedViews
<code>/thesis.pdf</code>	Text práce

Online zdroje

Projekty řešené v rámci této práce lze nalézt na těchto na adresách:

<https://github.com/PavelHryzlik/DiplomaThesis> - Text práce

<https://github.com/PavelHryzlik/ContractStandard> - Datový standard

<https://github.com/PavelHryzlik/TriadaEndpoint> - Konverzní modul

<https://github.com/PavelHryzlik/ContractViewer> - Webová aplikace

<http://opencontracts.azurewebsites.net/> - Testovací nasazení web. aplikace

D Příloha

```
1 @prefix : <http://tiny.cc/open-contracting#> .
2
3 @prefix com: <https://w3id.org/commerce#> .
4 @prefix dcterms: <http://purl.org/dc/terms/> .
5 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
6 @prefix gr: <http://purl.org/goodrelations/v1#> .
7 @prefix owl: <http://www.w3.org/2002/07/owl#> .
8 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
9 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
10 @prefix schema: <http://schema.org/> .
11 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
12 @prefix vann: <http://purl.org/vocab/vann/> .
13 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
14
15 ### — Metadata —
16
17 <http://purl.org/open-contracting/ontology> a owl:Ontology ;
18 owl:versionInfo "0.1" ;
19 dcterms:title "Ontologie pro smluvní údaje"@cs, "Contract ontology"@en ;
20 dcterms:description "Tento materiál navrhuje základní ontologii pro zveřejňová
21 ní smluv jako Linked Data nad rámec připravovaného zákona o registru smluv.
22 Cílem je sjednotit obdobné snahy úřadů o vyšší transparentci"@cs ;
23 dcterms:description "This material is proposing fundamental ontology for
24 disclosure contracts as Linked Data beyond the act of contracts register.
25 The aim is to consolidate similar efforts by the authorities of higher
26 transparency"@en ;
27
28 dcterms:modified "2015-11-30"^^xsd:date ;
29 vann:preferredNamespaceUri "http://tiny.cc/open-contracting#" ;
30 vann:preferredNamespacePrefix "cn" ;
31
32 dcterms:creator <http://www.ms.mff.cuni.cz/~hryzlik> ;
33 dcterms:rights <http://creativecommons.org/licenses/by/3.0/cz/> .
34
35 ### — Ontology author —
36
37 <http://www.ms.mff.cuni.cz/~hryzlik> a foaf:Person ;
38 foaf:name "Pavel Hryzlik" ;
39 foaf:mbox <mailto:hryzlik@gmail.com> .
40
41 ### — Classes —
42
43 :Document a owl:Class ;
44 rdfs:label "Dokument"@cs, "Document"@en ;
45 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
46
47 :Contract a owl:Class ;
48 rdfs:label "Smlouva"@cs, "Contract"@en ;
49 rdfs:subClassOf :Document ;
50 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
51
52 :Attachment a owl:Class ;
53 rdfs:label "Příloha"@cs, "Attachment"@en ;
54 rdfs:subClassOf :Document ;
55 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
56
57 :Amendment a owl:Class ;
58 rdfs:label "Dodatek"@cs, "Amendment"@en ;
59 rdfs:subClassOf :Document ;
60 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
61
62 :Order a owl:Class ;
63 rdfs:label "Objednávka"@cs, "Order"@en ;
64 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
65
66 :Invoice a owl:Class ;
67 rdfs:label "Faktura"@cs, "Invoice"@en ;
68 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
69
70 :Version a owl:Class ;
71 rdfs:label "Verze"@cs, "Version"@en ;
```

```

65   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
66
67   :Implementation a owl:Class ;
68     rdfs:label "Implementace"@cs, "Implementation"@en ;
69     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
70
71   :Milestone a owl:Class ;
72     rdfs:label "Milník"@cs, "Milestone"@en ;
73     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
74
75   com:Transaction vann:usageNote "Transakce"@cs, "Transaction"@en .
76
77   ### — Properties —
78
79   dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
80   dcterms:issued vann:usageNote "Zveřejněno"@cs, "Published"@en .
81   dcterms:language vann:usageNote "Jazyk"@cs, "Language"@en .
82
83   :documents a owl:ObjectProperty, owl:FunctionalProperty ;
84     rdfs:label "Dokumety"@cs, "Documents"@en ;
85     rdfs:range [ a rdfs:Class ; owl:unionOf ( :Contract :Attachment :Amendment ) ] ;
86     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
87
88   :orders a owl:ObjectProperty, owl:FunctionalProperty ;
89     rdfs:label "Objednávky"@cs, "Orders"@en ;
90     rdfs:range :Order ;
91     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
92
93   :invoices a owl:ObjectProperty, owl:FunctionalProperty ;
94     rdfs:label "Faktury"@cs, "Invoices"@en ;
95     rdfs:range :Invoice ;
96     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
97
98   :party a owl:ObjectProperty ;
99     rdfs:label "Smluvní strana"@cs, "Party"@en ;
100    rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Order :Invoice ) ] ;
101    rdfs:range gr:BusinessEntity ;
102    rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
103
104   :implementation a owl:ObjectProperty, owl:FunctionalProperty ;
105     rdfs:label "Rozšiřující entity"@cs, "Implementation"@en ;
106     rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Document :Order ) ] ;
107     rdfs:range :Implementation ;
108     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
109
110   :uri a owl:DatatypeProperty, owl:FunctionalProperty ;
111     rdfs:label "Uri"@cs, "Uri"@en ;
112     rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Attachment :Amendment
113       :Version ) ] ;
114     rdfs:range xsd:anyURI ;
115     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
116
117   :contract a owl:DatatypeProperty, owl:FunctionalProperty ;
118     rdfs:label "Smlouva"@cs, "Contract"@en ;
119     rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Attachment :Amendment ) ] ;
120     rdfs:range xsd:anyURI ;
121     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
122
123   :parentDocument a owl:DatatypeProperty, owl:FunctionalProperty ;
124     rdfs:label "Nadřazený dokument"@cs, "Parent document"@en ;
125     rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Order :Invoice ) ] ;
126     rdfs:range xsd:anyURI ;
127     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
128
129   :publisherId a owl:DatatypeProperty, owl:FunctionalProperty ;
130     rdfs:label "Id vydavatele"@cs, "Publisher Id"@en ;
131     rdfs:domain [ a rdfs:Class ; owl:unionOf ( com:Transaction :Version ) ] ;
132     rdfs:range xsd:string ;
133     rdfs:subPropertyOf dcterms:identifier ;
134     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
135
136   :dueDate a owl:DatatypeProperty, owl:FunctionalProperty ;

```

```

136   rdfs:label "Datum splatnosti"@cs, "Due date"@en ;
137   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Invoice :Milestone ) ] ;
138   rdfs:range xsd:dateTime ;
139   rdfs:subPropertyOf dcterms:date ;
140   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
141
142 :amount a owl:ObjectProperty ;
143   rdfs:label "Částka"@cs, "Amount"@en ;
144   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Order :Invoice
145     com:Transaction ) ] ;
145   rdfs:range gr:PriceSpecification ;
146   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
147
148 ### :Document properties —
149
150 schema:url vann:usageNote "Adresa URL fyzického umístění dokumentu"@cs, "The URL
151   of the physical location of the document"@en .
152 dcterms:type vann:usageNote "Typ dokumentu – Smlouva/Příloha/Dodatek"@cs, "
153   Document type – Contract/Attachment/Amendment"@en .
154
155 :valid a owl:DatatypeProperty, owl:FunctionalProperty ;
156   rdfs:label "Platnost"@cs, "Valid"@en ;
157   rdfs:domain :Document ;
158   rdfs:range xsd:boolean ;
159   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
160
161 :plainText a owl:DatatypeProperty, owl:FunctionalProperty ;
162   rdfs:label "Prostý text"@cs, "Plain text"@en ;
163   rdfs:domain :Document ;
164   rdfs:range xsd:string ;
165   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
166
167 :anonymised a owl:DatatypeProperty, owl:FunctionalProperty ;
168   rdfs:label "Anonymizováno"@cs, "Anonymised"@en ;
169   rdfs:domain :Document ;
170   rdfs:range xsd:boolean ;
171   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
172
173 :responsiblePerson a owl:ObjectProperty ;
174   rdfs:label "Zodpovědná osoba"@cs, "Responsible person"@en ;
175   rdfs:domain :Document ;
176   rdfs:range dcterms:Person ;
177   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
178
179 :publisher a owl:ObjectProperty ;
180   rdfs:label "Vydavatel"@cs, "Publisher"@en ;
181   rdfs:domain :Document ;
182   rdfs:range foaf:Organization ;
183   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
184
185 :address a owl:ObjectProperty ;
186   rdfs:label "Adresa"@cs, "Address"@en ;
187   rdfs:domain :Document ;
188   rdfs:range schema:PostalAddress ;
189   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
190
191 :version a owl:ObjectProperty ;
192   rdfs:label "Verze"@cs, "Version"@en ;
193   rdfs:domain :Document ;
194   rdfs:range :Version ;
195   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
196
197 ### :Contract properties —
198
199 dcterms:title vann:usageNote "Předmět smlouvy"@cs, "Object of the contract"@en .
200 dcterms:created vann:usageNote "Datum podpisu"@cs, "Signed date"@en .
201 dcterms:description vann:usageNote "Popis předmětu smlouvy"@cs, "Description of
202   object of the contract"@en .
203
204 :awardID a owl:DatatypeProperty, owl:FunctionalProperty ;
205   rdfs:label "Evidenční číslo veřejné zakázky"@cs, "Award public contract Id"@en
206   ;
207   rdfs:domain :Contract ;

```

```

204   rdfs:range xsd:string ;
205   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
206
207 :awardProfileID a owl:DatatypeProperty , owl:FunctionalProperty ;
208   rdfs:label "Číslo zakázky na profilu zadavatele"@cs, "Award public contract
209   profile Id"@en ;
210   rdfs:domain :Contract ;
211   rdfs:range xsd:string ;
212   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
213
214 :contractType a owl:DatatypeProperty , owl:FunctionalProperty ;
215   rdfs:label "Typ smlouvy"@cs, "Contract type"@en ;
216   rdfs:domain :Contract ;
217   rdfs:range xsd:string ;
218   rdfs:subPropertyOf dcterms:type ;
219   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
220
221 :subjectType a owl:DatatypeProperty , owl:FunctionalProperty ;
222   rdfs:label "Typ zboží/služeb"@cs, "Types of goods / services"@en ;
223   rdfs:domain :Contract ;
224   rdfs:range xsd:string ;
225   rdfs:subPropertyOf dcterms:type ;
226   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
227
228 :amountNoVat a owl:ObjectProperty ;
229   rdfs:label "Cena bez DPH"@cs, "Price without VAT"@en ;
230   rdfs:domain :Contract ;
231   rdfs:range gr:PriceSpecification ;
232   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
233
234 :priceAnnual a owl:DatatypeProperty , owl:FunctionalProperty ;
235   rdfs:label "Identifikuje, pokud je v Amount roční částka"@cs, "Annual price "
236   @en ;
237   rdfs:domain :Contract ;
238   rdfs:range xsd:boolean ;
239   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
240
241 :validFrom a owl:DatatypeProperty , owl:FunctionalProperty ;
242   rdfs:label "Datum účinnosti smlouvy"@cs, "Effective date of the contract"@en .
243   rdfs:domain :Contract ;
244   rdfs:range xsd:dateTime ;
245   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
246
247 :validUntil a owl:DatatypeProperty , owl:FunctionalProperty ;
248   rdfs:label "Datum ukončení účinnosti smlouvy"@cs, "Expiration date of the
249   contract"@en .
250   rdfs:domain :Contract ;
251   rdfs:range xsd:dateTime ;
252   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
253
254 :funding a owl:DatatypeProperty , owl:FunctionalProperty ;
255   rdfs:label "Převažující financování"@cs, "Funding"@en ;
256   rdfs:domain :Contract ;
257   rdfs:range xsd:string ;
258   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
259
260 :attachment a owl:ObjectProperty , owl:FunctionalProperty ;
261   rdfs:label "Příloha"@cs, "Attachment"@en ;
262   rdfs:domain :Contract ;
263   rdfs:range :Attachment ;
264   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
265
266 :amendment a owl:ObjectProperty , owl:FunctionalProperty ;
267   rdfs:label "Dodatek"@cs, "Amendment"@en ;
268   rdfs:domain :Contract ;
269   rdfs:range :Amendment ;
270   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
271
272 :currentValidContract a owl:DatatypeProperty , owl:FunctionalProperty ;
273   rdfs:label "Aktuálně platné znění smlouvy"@cs, "Currently valid wording of the
274   contract"@en ;
275   rdfs:domain :Contract ;
276   rdfs:range xsd:anyURI ;

```

```

273   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
274
275   :competency a owl:DatatypeProperty ;
276     rdfs:label "Indikuje, zda-li se jedná o soukromoprávní nebo veřejnoprávní
smlouvu"@cs, "Private or public contract"@en ;
277     rdfs:domain :Contract ;
278     rdfs:range xsd:string ;
279     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
280
281   ### :Version properties —
282
283   dcterms:issued vann:usageNote "Zveřejněno"@cs, "Published"@en .
284
285   :versionOrder a owl:DatatypeProperty, owl:FunctionalProperty ;
286     rdfs:label "Pořadí verzí"@cs, "Order version"@en ;
287     rdfs:domain :Version ;
288     rdfs:range xsd:integer ;
289     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
290
291   ### :Implementation properties —
292
293   :milestone a owl:ObjectProperty, owl:FunctionalProperty ;
294     rdfs:label "Mílník"@cs, "Milestone"@en ;
295     rdfs:domain :Implementation ;
296     rdfs:range :Milestone ;
297     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
298
299   :transaction a owl:ObjectProperty, owl:FunctionalProperty ;
300     rdfs:label "Transakce"@cs, "Transaction"@en ;
301     rdfs:domain :Implementation ;
302     rdfs:range com:Transaction ;
303     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
304
305   ### :Milestone properties —
306
307   dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
308
309   ### com:Transaction properties —
310
311   dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
312   dcterms:date vann:usageNote "Datum a čas proběhlé transakce"@cs, "Date and time
of the transaction took place"@en .
313   com:source vann:usageNote "Informace o odesílateli"@cs, "Information about the
sender"@en .
314   com:destination vann:usageNote "Informace o příjemci"@cs, "Recipient Information
"@en .
315
316   ### :Attachment properties —
317
318   dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
319   dcterms:title vann:usageNote "Název"@cs, "Name"@en .
320   dcterms:created vann:usageNote "Datum podpisu"@cs, "Date of signature"@en .
321
322   :attachmentOrder a owl:DatatypeProperty, owl:FunctionalProperty ;
323     rdfs:label "Pořadové číslo přílohy"@cs, "Serial number of the attachment"@en ;
324     rdfs:domain :Attachment ;
325     rdfs:range xsd:integer ;
326     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
327
328   ### :Amendment properties —
329
330   dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
331   dcterms:title vann:usageNote "Název"@cs, "Name"@en .
332   dcterms:created vann:usageNote "Datum podpisu"@cs, "Date of signature"@en .
333
334   :amendmentOrder a owl:DatatypeProperty, owl:FunctionalProperty ;
335     rdfs:label "Pořadové číslo dodatku"@cs, "Serial number of the amendment"@en ;
336     rdfs:domain :Amendment ;
337     rdfs:range xsd:integer ;
338     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
339
340   ### :Order properties —
341

```

```

342 dcterms:title vann:usageNote ""@cs, ""@en .
343 dcterms:created vann:usageNote "Datum podpisu"@cs, "Date of signature"@en .
344
345 ### :Invoice properties —
346
347 dcterms:title vann:usageNote "Název"@cs, "Name"@en .
348 dcterms:created vann:usageNote "Datum podpisu"@cs, "Date of signature"@en .
349
350 ### gr:BusinessEntity properties —
351
352 dcterms:identifier vann:usageNote "ID"@cs, "ID"@en .
353 gr:legalName vann:usageNote "Název"@cs, "Name"@en .
354 schema:addressCountry vann:usageNote "Země původu, 3-písmený ISO kód"@cs, "
    Country of origin, 3-letter ISO code"@en .
355
356 :localID a owl:DatatypeProperty, owl:FunctionalProperty ;
357   rdfs:label "Jednoznačný identifikátor"@cs, "Unique identifier"@en ;
358   rdfs:domain gr:BusinessEntity ;
359   rdfs:range xsd:integer ;
360   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
361
362 :payer a owl:DatatypeProperty, owl:FunctionalProperty ;
363   rdfs:label "Plátce"@cs, "Payer"@en ;
364   rdfs:domain gr:BusinessEntity ;
365   rdfs:range xsd:boolean ;
366   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
367
368 :noID a owl:DatatypeProperty, owl:FunctionalProperty ;
369   rdfs:label "Indikuje že subjekt nemá IČ, nebo zahraniční ID"@cs, "Without
    company identification number"@en ;
370   rdfs:domain gr:BusinessEntity ;
371   rdfs:range xsd:boolean ;
372   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
373
374 :paysVAT a owl:DatatypeProperty, owl:FunctionalProperty ;
375   rdfs:label vann:usageNote "Plátce DPH"@cs, "VAT payer"@en ;
376   rdfs:domain gr:BusinessEntity ;
377   rdfs:range xsd:boolean ;
378   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
379
380 :superiorInstitution a owl:ObjectProperty ;
381   rdfs:label "Nadřazená instituce"@cs, "Parent institution"@en ;
382   rdfs:domain gr:BusinessEntity ;
383   rdfs:range gr:BusinessEntity ;
384   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
385
386 ### foaf:Organization properties —
387
388 :authentication a owl:DatatypeProperty, owl:FunctionalProperty ;
389   rdfs:label "Značí stupeň ověření zveřejňující strany"@cs, "Verification
    degree"@en ;
390   rdfs:domain foaf:Organization ;
391   rdfs:range xsd:string ;
392   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
393
394 ### schema:PostalAddress properties —
395
396 :nuts a owl:DatatypeProperty, owl:FunctionalProperty ;
397   rdfs:label "Normalizovaná klasifikace územních celků"@cs, "Nomenclature of
    territorial units"@en ;
398   rdfs:domain schema:PostalAddress ;
399   rdfs:range xsd:string ;
400   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .

```

RDF Ontologie pro smlouvy

E Příloha

```
1 # The R2RML mapping contracts
2
3 @base      <http://tiny.cc/open-contracting#> .
4 @prefix rr: <http://www.w3.org/ns/r2rml#> .
5
6 @prefix cn: <http://tiny.cc/open-contracting#> .
7 @prefix com: <https://w3id.org/commerce#> .
8 @prefix dc: <http://purl.org/dc/terms/> .
9 @prefix dcmi: <http://dublincore.org/documents/dcmi-type-vocabulary/> .
10 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
11 @prefix gr: <http://purl.org/goodrelations/v1#> .
12 @prefix owl: <http://www.w3.org/2002/07/owl#> .
13 @prefix pc: <http://purl.org/procurement/public-contracts#> .
14 @prefix schema: <http://schema.org/> .
15 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
16 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
17 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
18
19 <#PublisherTableView> rr:sqlQuery """
20 SELECT NAZEVORGANIZACE, ICO,
21 IIF(ICO is NULL, 'true', 'false') As NoID
22 FROM [triada].[TRLOGADR] WHERE HLAVNI = 'T'
23 """ .
24
25 <#ContractsTableView> rr:sqlQuery """
26 SELECT Smlouva.ID,
27 Verze.PORADIVERZE, Verze.PREDMET, Verze.POPIS_POPIS, Verze.TYPSMLOUVY, Mena.
28 ZKRATKA, Verze.CELKOVACASTKA,
29 Verze.DATUMPODPISU, Verze.DATUMUCINOSTI, Verze.DATUMKONCENI, Verze.
30 SMLUVSTRANROZD, Verze.DATUMZMENYSTAVULIS,
31 VZakazka.EVIDENCNISLOZAKAZKY, VZakazka.EVIDENCNISLOFORMULARE,
32 (CASE Verze.ANONYMIZOVANO
33 WHEN 'T' THEN 'true'
34 WHEN 'F' THEN 'false'
35 END) AS Anonymizovano
36 FROM [triada].[ESMLUV.SMLOUVA] AS Smlouva
37 JOIN [triada].[ESMLUV.VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
38 JOIN [triada].[ESMLUV.MENA] As Mena ON Verze.MENA = Mena.ID
39 LEFT JOIN [triada].[ESMLUV.VERZAKAZKA] As VZakazka ON Verze.VEREJNAZAKAZKA =
40 VZakazka.ID
41 WHERE Smlouva.RODIC is NULL
42 """ .
43
44 <#ContractTypesTableView> rr:sqlQuery """
45 SELECT Smlouva.ID, Verze.PORADIVERZE,
46 (CASE TypSmlouvy.TYP
47 WHEN '1' THEN 'Nájemní smlouva'
48 WHEN '2' THEN 'Darovací smlouva'
49 WHEN '3' THEN 'Kupní smlouva'
50 WHEN '4' THEN 'Směnná smlouva'
51 WHEN '5' THEN 'Pojistná smlouva'
52 WHEN '6' THEN 'Smlouva o výpůjčce'
53 WHEN '7' THEN 'Smlouva o dílo'
54 WHEN '8' THEN 'Licenční smlouva'
55 WHEN '9' THEN 'Mandátní smlouva'
56 WHEN '10' THEN 'Leasingová smlouva'
57 WHEN '11' THEN 'Pachtovní smlouva'
58 WHEN '12' THEN 'Smlouva o zřízení věcného břemene'
59 WHEN '13' THEN 'Smlouva o provedení stavby'
60 WHEN '14' THEN 'Smlouva o provedení práce'
61 WHEN '15' THEN 'Smlouva o provedení uměleckého výkonu'
62 WHEN '16' THEN 'Smlouva o úvěru'
63 WHEN '17' THEN 'Smlouva o uzavření budoucí smlouvy'
64 WHEN '18' THEN 'Veřejnoprávní smlouva'
65 WHEN '19' THEN 'Jiná'
66 END) AS Typ,
67 Iif(TypSmlouvy.TYP = '18', 'Veřejnoprávní smlouva', 'Soukromoprávní smlouva')
68 As Kompetence
69 FROM [triada].[ESMLUV.SMLOUVA] AS Smlouva
```



```

66 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
67 JOIN [triada].[ESMLUV_TYPSMLOUVY] As TypSmlouvy ON Verze.TYPSMLOUVY =
    TypSmlouvy.ID
68 WHERE Smlouva.RODIC is NULL
69 """
70
71 <#PartiesTableView> rr:sqlQuery """
72 SELECT Smlouva.ID, Verze.PORADIVERZE, SmlStrana.HAD_POUZITA
73 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
74 JOIN [triada].[ESMLUV_SMLUVSTRANA] AS SmlStrana ON Verze.SMLUVSTRANROZD =
    SmlStrana.SMLUVSTRANYROZDELOVNIK
75 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
76 WHERE Smlouva.RODIC is NULL
77 """
78
79 <#PartyTableView> rr:sqlQuery """
80 SELECT SmlStrana1.HAD_POUZITA,
81 Had.NAZEVSUBJEKTU, Had.ICO, Had.STAT, Had.ULICE, Had.CISLA, Had.MESTO, Had.
    PSC,
82 IIF(Had.ICO is NULL, 'true', 'false') As NoID,
83 (CASE SmlStrana1.PLATCEDPH
84     WHEN 'T' THEN 'true'
85     WHEN 'F' THEN 'false'
86 END) AS PlatceDPH
87 FROM [triada].[ESMLUV_SMLUVSTRANA] AS SmlStrana1
88 JOIN
89 (SELECT DISTINCT HAD_POUZITA, MIN(ID) AS MinId
90 FROM [triada].[ESMLUV_SMLUVSTRANA]
91 GROUP BY HAD_POUZITA) AS SmlStrana2
92 ON SmlStrana1.ID = SmlStrana2.MinId
93 JOIN [triada].[HAD_POUZITA] AS Had
94 ON SmlStrana2.HAD_POUZITA = Had.ID_POUZITA
95 """
96
97 <#ContractValidTableView> rr:sqlQuery """
98 SELECT Smlouva.ID, Verze.PORADIVERZE,
99 IIF(Smlouva.AKTUALNIVERZE = Verze.ID, 'true', 'false') As Valid
100 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva JOIN [triada].[ESMLUV_VERZESMLOUVY]
    As Verze ON Smlouva.ID = Verze.SMLOUVA
101 WHERE Smlouva.RODIC is NULL
102 """
103
104 <#ContractFilesTableView> rr:sqlQuery """
105 SELECT Smlouva.ID, Verze.PORADIVERZE, Soubor.NAZEVSOUBORU, Soubor .
    SADADUL_ULOZISTEID
106 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
107 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] AS Soubor ON Verze.SOUBOR = Soubor.ID
108 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
109 WHERE Smlouva.RODIC is NULL
110 """
111
112 <#ContractResponsiblePersons1TableView> rr:sqlQuery """
113 SELECT Smlouva.ID, Verze.PORADIVERZE,
114 LTRIM(CONCAT(
115 ISNULL(Uzivatel.TITUL_PRED, ''), ' ',
116 ISNULL(Uzivatel.JMENO, ''), ' ',
117 ISNULL(Uzivatel.PRIJMENI, ''), ' ',
118 ISNULL(Uzivatel.TITUL_ZA, ''))) AS CeleJmeno
119 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
120 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID = ExtKontakt .
    VERZESMLOUVY
121 JOIN [triada].[TRLUZIVATEL] AS Uzivatel ON Uzivatel.CISLO = ExtKontakt .
    UZIVATEL
122 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
123 WHERE Smlouva.RODIC is NULL
124 """
125
126 <#ContractResponsiblePersons2TableView> rr:sqlQuery """
127 SELECT Smlouva.ID, Verze.PORADIVERZE,
128 LTRIM(CONCAT(
129 ISNULL(ExtKontakt.TITUL_PRED, ''), ' ',
130 ISNULL(ExtKontakt.JMENO, ''), ' ',
131 ISNULL(ExtKontakt.PRIJMENI, ''), ' ',

```

```

132 ISNULL(ExtKontakt.TITULZA, '')) AS CeleJmeno
133 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
134 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID = ExtKontakt.
    VERZESMLOUVY
135 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
136 WHERE ExtKontakt.UZIVATEL IS NULL AND Smlouva.RODIC is NULL
137 """
138
139 <#AmendmentTableView> rr:sqlQuery """
140 SELECT Dodatek.ID, Dodatek.RODIC, Verze.PORADIVERZE,
141 VerzeDodatku.PORADIVERZE As PoradiVerzeDodatku, VerzeDodatku.PREDMET,
    VerzeDodatku.POPIS_POPIS,
142 VerzeDodatku.DATUMPODPISU, VerzeDodatku.DATUMZMENYSTAVUTS,
143 (CASE VerzeDodatku.ANONYMIZOVANO
144     WHEN 'T' THEN 'true'
145     WHEN 'F' THEN 'false'
146 END) AS Anonymizovano
147 FROM [triada].[ESMLUV_SMLOUVA] AS Dodatek
148 JOIN [triada].[ESMLUV_VERZESMLOUVY] As VerzeDodatku ON Dodatek.ID =
    VerzeDodatku.SMLOUVA
149 JOIN [triada].[ESMLUV_SMLOUVA] As Smlouva ON Dodatek.RODIC = Smlouva.ID
150 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
151 WHERE Dodatek.RODIC is not NULL
152 """
153
154 <#AmendmentValidTableView> rr:sqlQuery """
155 SELECT Smlouva.ID, Verze.PORADIVERZE,
156 IIF(Smlouva.AKTUALNIVERZE = Verze.ID, 'true', 'false') As Valid
157 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva JOIN [triada].[ESMLUV_VERZESMLOUVY]
    As Verze ON Smlouva.ID = Verze.SMLOUVA
158 WHERE Smlouva.RODIC is not NULL
159 """
160
161 <#AmendmentFilesTableView> rr:sqlQuery """
162 SELECT Smlouva.ID, Verze.PORADIVERZE, Soubor.NAZEVSOUBORU, Soubor.
    SADADULULOZISTEID
163 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
164 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] AS Soubor ON Verze.SOUBOR = Soubor.ID
165 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
166 WHERE Smlouva.RODIC is not NULL
167 """
168
169 <#AmendmentResponsiblePersons1TableView> rr:sqlQuery """
170 SELECT Smlouva.ID, Verze.PORADIVERZE,
171 LTRIM(CONCAT(
172 ISNULL(Uzivatel.TITUL_PRED, ''), ' ',
173 ISNULL(Uzivatel.JMENO, ''), ' ',
174 ISNULL(Uzivatel.PRIJMENI, ''), ' ',
175 ISNULL(Uzivatel.TITUL_ZA, ''))) AS CeleJmeno
176 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
177 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID = ExtKontakt.
    VERZESMLOUVY
178 JOIN [triada].[TRL_UZIVATEL] AS Uzivatel ON Uzivatel.CISLO = ExtKontakt.
    UZIVATEL
179 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
180 WHERE Smlouva.RODIC is not NULL
181 """
182
183 <#AmendmentResponsiblePersons2TableView> rr:sqlQuery """
184 SELECT Smlouva.ID, Verze.PORADIVERZE,
185 LTRIM(CONCAT(
186 ISNULL(ExtKontakt.TITUL_PRED, ''), ' ',
187 ISNULL(ExtKontakt.JMENO, ''), ' ',
188 ISNULL(ExtKontakt.PRIJMENI, ''), ' ',
189 ISNULL(ExtKontakt.TITUL_ZA, ''))) AS CeleJmeno
190 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
191 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID = ExtKontakt.
    VERZESMLOUVY
192 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
193 WHERE ExtKontakt.UZIVATEL IS NULL AND Smlouva.RODIC is not NULL
194 """
195
196 <#AttachmentTableView> rr:sqlQuery """

```

```

197 SELECT DISTINCT Priloha.ID, Priloha.POPIS_NAZEV, Priloha.NAZEVSOUBORU,
198 Priloha.SADADUL_ULOZISTEID, Priloha.OKAMZIKVYTVORENI,
199 (CASE Verze.ANONYMIZOVANO
200     WHEN 'T' THEN 'true'
201     WHEN 'F' THEN 'false'
202 END) AS Anonymizovano
203 FROM [triada].[ESMLUV_VERZESMLOUVY] As Verze
204 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] As Priloha ON Verze.SOUBOR = Priloha.
    RODIC
205 """ .
206
207 <#AttachmentToContractTableView> rr:sqlQuery """
208 SELECT Smlouva.ID As SmlouvaID, Priloha.ID, Verze.PORADIVERZE
209 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
210 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
211 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] As Priloha ON Verze.SOUBOR = Priloha.
    RODIC
212 """ .
213
214 <#MilestoneTableView> rr:sqlQuery """
215 SELECT Smlouva.ID, Verze.PORADIVERZE,
216 Milnik.ID As MilestoneID, Milnik.NAZEV, Milnik.DATUMUCINOSTIML
217 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
218 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
219 JOIN [triada].[ESMLUV_MILNIK] As Milnik ON Verze.ID = Milnik.VERZESMLOUVY
220 """ .
221
222 <#Publisher> a rr:TriplesMap;
223   rr:logicalTable <#PublisherTableView>;
224   rr:subjectMap [
225     rr:template "http://localhost:7598/publisher";
226     rr:class foaf:Organization;
227   ];
228   rr:predicateObjectMap [
229     rr:predicate gr:legalName;
230     rr:objectMap [ rr:column "NAZEVORGANIZACE" ];
231   ];
232   rr:predicateObjectMap [
233     rr:predicate cn:noID;
234     rr:objectMap [
235       rr:column "[NoID]";
236     ];
237     rr:datatype xsd:boolean;
238   ];
239   rr:predicateObjectMap [
240     rr:predicate dc:identifier;
241     rr:objectMap [ rr:column "[ICO]" ];
242   ];
243   rr:predicateObjectMap [
244     rr:predicate owl:sameAs;
245     rr:objectMap [ rr:template "http://linked.opendata.cz/resource/business-
    entity/CZ{ICO}" ];
246   ];
247   rr:predicateObjectMap [
248     rr:predicate schema:addressCountry;
249     rr:objectMap [ rr:constant "CZE" ];
250   ].
251
252 <#Contract> a rr:TriplesMap;
253   rr:logicalTable <#ContractsTableView>;
254   rr:subjectMap [
255     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}";
256     rr:class cn:Contract;
257   ];
258   rr:predicateObjectMap [
259     rr:predicate dcmi:type;
260     rr:objectMap [ rr:constant "Smlouva" ];
261   ];
262   rr:predicateObjectMap [
263     rr:predicate cn:anonymised;
264     rr:objectMap [
265       rr:column "[Anonymizovano]";
266     ];
    rr:datatype xsd:boolean;

```

```

267     ];
268 ];
269 rr:predicateObjectMap [
270   rr:predicate dc:title;
271   rr:objectMap [ rr:column "[PREDMET]" ];
272 ];
273 rr:predicateObjectMap [
274   rr:predicate dc:description;
275   rr:objectMap [ rr:column "[POPIS_POPIS]" ];
276 ];
277 rr:predicateObjectMap [
278   rr:predicate dc:created;
279   rr:objectMap [
280     rr:column "[DATUMPODPISU]";
281     rr:datatype xsd:date;
282   ];
283 ];
284 rr:predicateObjectMap [
285   rr:predicate cn:validFrom;
286   rr:objectMap [
287     rr:column "[DATUMUCINOSTI]";
288     rr:datatype xsd:date;
289   ];
290 ];
291 rr:predicateObjectMap [
292   rr:predicate cn:validUntil;
293   rr:objectMap [
294     rr:column "[DATUMUKONCENI]";
295     rr:datatype xsd:date;
296   ];
297 ];
298 rr:predicateObjectMap [
299   rr:predicate cn:priceAnnual;
300   rr:objectMap [
301     rr:template "false";
302     rr:datatype xsd:boolean;
303   ];
304 ];
305 rr:predicateObjectMap [
306   rr:predicate cn:funding;
307   rr:objectMap [ rr:constant "vlastní" ];
308 ];
309 rr:predicateObjectMap [
310   rr:predicate cn:awardID;
311   rr:objectMap [ rr:column "[EVIDENCNICISLOZAKAZKY]" ];
312 ];
313 rr:predicateObjectMap [
314   rr:predicate pc:publicContract;
315   rr:objectMap [ rr:template "http://linked.opendata.cz/resource/domain/
316     buyer-profiles/contract/cz/{EVIDENCNICISLOZAKAZKY}" ];
317 ];
318 rr:predicateObjectMap [
319   rr:predicate cn:awardProfileID;
320   rr:objectMap [ rr:column "[EVIDENCNICISLOFORMULARE]" ];
321 ];
322 rr:predicateObjectMap [
323   rr:predicate cn:amount;
324   rr:objectMap [ rr:template "http://localhost:7598/contract/{ID}/{
325     PORADIVERZE}/amount" ];
326 ];
327 rr:predicateObjectMap [
328   rr:predicate dc:publisher;
329   rr:objectMap [ rr:template "http://localhost:7598/publisher" ];
330 ];
331 rr:predicateObjectMap [
332   rr:predicate cn:version;
333   rr:objectMap [ rr:template "http://localhost:7598/contract/{ID}/{
334     PORADIVERZE}/version" ];
335 ];
336 <#ContractPriceSpecification> a rr:TriplesMap;
337   rr:logicalTable <#ContractsTableView>;
338   rr:subjectMap [

```

```

337     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}/amount";
338     rr:class gr:PriceSpecification;
339 ];
340 rr:predicateObjectMap [
341     rr:predicate gr:hasCurrencyValue;
342     rr:objectMap [
343         rr:column "[CELKOVACASTKA]";
344         rr:datatype xsd:float;
345     ];
346 ];
347 rr:predicateObjectMap [
348     rr:predicate gr:hasCurrency;
349     rr:objectMap [ rr:column "[ZKRATKA]" ];
350 ].
351
352 <#ContractType> a rr:TriplesMap;
353     rr:logicalTable <#ContractTypesTableView>;
354     rr:subjectMap [
355         rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
356     ];
357     rr:predicateObjectMap [
358         rr:predicate cn:contractType;
359         rr:objectMap [ rr:column "[Typ]" ];
360     ];
361     rr:predicateObjectMap [
362         rr:predicate cn:competency;
363         rr:objectMap [ rr:column "[Kompetence]" ];
364     ].
365
366 <#ContractValid> a rr:TriplesMap;
367     rr:logicalTable <#ContractValidTableView>;
368     rr:subjectMap [
369         rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
370     ];
371     rr:predicateObjectMap [
372         rr:predicate cn:valid;
373         rr:objectMap [
374             rr:column "[Valid]";
375             rr:datatype xsd:boolean;
376         ];
377     ].
378
379 <#ContractFile> a rr:TriplesMap;
380     rr:logicalTable <#ContractFilesTableView>;
381     rr:subjectMap [
382         rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
383     ];
384     rr:predicateObjectMap [
385         rr:predicate schema:url;
386         rr:objectMap [ rr:template "http://localhost:7598/file/{
387             SADADULULOZISTEID}/{NAZEVSOUBORU}" ];
388     ].
389
390 <#ContractResponsiblePerson1> a rr:TriplesMap;
391     rr:logicalTable <#ContractResponsiblePersons1TableView>;
392     rr:subjectMap [
393         rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
394     ];
395     rr:predicateObjectMap [
396         rr:predicate cn:responsiblePerson;
397         rr:objectMap [ rr:column "[CeleJmeno]" ];
398     ].
399
400 <#ContractResponsiblePerson2> a rr:TriplesMap;
401     rr:logicalTable <#ContractResponsiblePersons2TableView>;
402     rr:subjectMap [
403         rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
404     ];
405     rr:predicateObjectMap [
406         rr:predicate cn:responsiblePerson;
407         rr:objectMap [ rr:column "[CeleJmeno]" ];
408     ].

```

```

409 <#ContractVersion> a rr:TriplesMap;
410   rr:logicalTable <#ContractsTableView>;
411   rr:subjectMap [
412     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}/version";
413     rr:class cn:Version;
414   ];
415   rr:predicateObjectMap [
416     rr:predicate cn:uri;
417     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}" ];
418   ];
419   rr:predicateObjectMap [
420     rr:predicate cn:versionOrder;
421     rr:objectMap [
422       rr:column "[PORADIVERZE]";
423       rr:datatype xsd:integer;
424     ];
425   ];
426   rr:predicateObjectMap [
427     rr:predicate dc:issued;
428     rr:objectMap [
429       rr:column "[DATUMZMENYSTAVUTS]";
430       rr:datatype xsd:dateTime;
431     ];
432   ].
433
434 <#Parties> a rr:TriplesMap;
435   rr:logicalTable <#PartiesTableView>;
436   rr:subjectMap [
437     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}"
438   ];
439   rr:predicateObjectMap [
440     rr:predicate cn:party;
441     rr:objectMap [ rr:template "http://localhost:7598/party/{HAD_POUZITA}"
442   ];
443
444 <#Party> a rr:TriplesMap;
445   rr:logicalTable <#PartyTableView>;
446   rr:subjectMap [
447     rr:template "http://localhost:7598/party/{HAD_POUZITA}";
448     rr:class gr:BusinessEntity;
449   ];
450   rr:predicateObjectMap [
451     rr:predicate gr:legalName;
452     rr:objectMap [ rr:column "[NAZEV.SUBJEKTU]" ];
453   ];
454   rr:predicateObjectMap [
455     rr:predicate dc:identifier;
456     rr:objectMap [ rr:column "[ICO]" ];
457   ];
458   rr:predicateObjectMap [
459     rr:predicate owl:sameAs;
460     rr:objectMap [ rr:template "http://linked.opendata.cz/resource/business-
entity/CZ{ICO}" ];
461   ];
462   rr:predicateObjectMap [
463     rr:predicate cn:noID;
464     rr:objectMap [
465       rr:column "[NoID]";
466       rr:datatype xsd:boolean;
467     ];
468   ];
469   rr:predicateObjectMap [
470     rr:predicate cn:localID;
471     rr:objectMap [
472       rr:column "[HAD_POUZITA]";
473       rr:datatype xsd:integer;
474     ];
475   ];
476   rr:predicateObjectMap [
477     rr:predicate schema:address;

```

```

478     rr:objectMap [ rr:template "http://localhost:7598/party/{HAD_POUZITA}/
address" ];
479 ];
480 rr:predicateObjectMap [
481     rr:predicate schema:addressCountry;
482     rr:objectMap [ rr:column "[STAT]" ];
483 ];
484 rr:predicateObjectMap [
485     rr:predicate cn:paysVAT;
486     rr:objectMap [
487         rr:column "[PlatceDPH]";
488         rr:datatype xsd:boolean;
489     ];
490 ];
491
492 <#Address> a rr:TriplesMap;
493 rr:logicalTable <#PartyTableView>;
494 rr:subjectMap [
495     rr:template "http://localhost:7598/party/{HAD_POUZITA}/address";
496     rr:class schema:PostalAddress;
497 ];
498 rr:predicateObjectMap [
499     rr:predicate schema:streetAddress;
500     rr:objectMap [
501         rr:template "{ULICE} {CISLA}";
502         rr:termType rr:Literal;
503     ];
504 ];
505 rr:predicateObjectMap [
506     rr:predicate schema:postalCode;
507     rr:objectMap [
508         rr:column "[PSC]";
509         rr:datatype xsd:integer;
510     ];
511 ];
512 rr:predicateObjectMap [
513     rr:predicate schema:addressLocality;
514     rr:objectMap [ rr:column "[MESTO]" ];
515 ];
516
517 <#ContractAmendments> a rr:TriplesMap;
518 rr:logicalTable <#AmendmentTableView>;
519 rr:subjectMap [
520     rr:template "http://localhost:7598/contract/{RODIC}/{PORADIVERZE}"
521 ];
522 rr:predicateObjectMap [
523     rr:predicate cn:amendment;
524     rr:objectMap [ rr:template "http://localhost:7598/amendment/{ID}/{
PoradiVerzeDodatku}" ];
525 ];
526
527 <#ContractAttachments> a rr:TriplesMap;
528 rr:logicalTable <#AttachmentToContractTableView>;
529 rr:subjectMap [
530     rr:template "http://localhost:7598/contract/{SmlouvaID}/{PORADIVERZE}"
531 ];
532 rr:predicateObjectMap [
533     rr:predicate cn:attachment;
534     rr:objectMap [ rr:template "http://localhost:7598/attachment/{ID}/1" ];
535 ];
536
537 <#Amendment> a rr:TriplesMap;
538 rr:logicalTable <#AmendmentTableView>;
539 rr:subjectMap [
540     rr:template "http://localhost:7598/amendment/{ID}/{PoradiVerzeDodatku}";
541     rr:class cn:Amendment;
542 ];
543 rr:predicateObjectMap [
544     rr:predicate dc:isTypeOf;
545     rr:objectMap [ rr:constant "Dodatek" ];
546 ];
547 rr:predicateObjectMap [
548     rr:predicate cn:anonymised;

```

```

549     rr:objectMap [
550         rr:column "[Anonymizovano]";
551         rr:datatype xsd:boolean;
552     ];
553 ];
554 rr:predicateObjectMap [
555     rr:predicate dc:title;
556     rr:objectMap [ rr:column "[PREDMET]" ];
557 ];
558 rr:predicateObjectMap [
559     rr:predicate cn:contract;
560     rr:objectMap [ rr:template "http://localhost:7598/contract/{RODIC}/{
PORADIVERZE}" ];
561 ];
562 rr:predicateObjectMap [
563     rr:predicate dc:identifier;
564     rr:objectMap [ rr:column "[ID]" ];
565 ];
566 rr:predicateObjectMap [
567     rr:predicate dc:created;
568     rr:objectMap [
569         rr:column "[DATUMPODPISU]";
570         rr:datatype xsd:date;
571     ];
572 ];
573 rr:predicateObjectMap [
574     rr:predicate dc:publisher;
575     rr:objectMap [ rr:template "http://localhost:7598/publisher" ];
576 ];
577 rr:predicateObjectMap [
578     rr:predicate cn:version;
579     rr:objectMap [ rr:template "http://localhost:7598/amendment/{ID}/{
PoradiVerzeDodatku}/version" ];
580 ];
581
582 <#AmendmentValid> a rr:TriplesMap;
583     rr:logicalTable <#AmendmentValidTableView>;
584     rr:subjectMap [
585         rr:template "http://localhost:7598/amendment/{ID}/{PORADIVERZE}"
586     ];
587     rr:predicateObjectMap [
588         rr:predicate cn:valid;
589         rr:objectMap [
590             rr:column "[Valid]";
591             rr:datatype xsd:boolean;
592         ];
593     ];
594
595 <#AmendmentFile> a rr:TriplesMap;
596     rr:logicalTable <#AmendmentFilesTableView>;
597     rr:subjectMap [
598         rr:template "http://localhost:7598/amendment/{ID}/{PORADIVERZE}"
599     ];
600     rr:predicateObjectMap [
601         rr:predicate schema:url;
602         rr:objectMap [ rr:template "http://localhost:7598/file/{
SADADULULOZISTEID}/{NAZEVSOUBORU}" ];
603     ];
604
605 <#AmendmentResponsiblePerson1> a rr:TriplesMap;
606     rr:logicalTable <#AmendmentResponsiblePersons1TableView>;
607     rr:subjectMap [
608         rr:template "http://localhost:7598/amendment/{ID}/{PORADIVERZE}"
609     ];
610     rr:predicateObjectMap [
611         rr:predicate cn:responsiblePerson;
612         rr:objectMap [ rr:column "[CeleJmeno]" ];
613     ];
614
615 <#AmendmentResponsiblePerson2> a rr:TriplesMap;
616     rr:logicalTable <#AmendmentResponsiblePersons2TableView>;
617     rr:subjectMap [
618         rr:template "http://localhost:7598/amendment/{ID}/{PORADIVERZE}"

```



```

619 ];
620 rr:predicateObjectMap [
621     rr:predicate cn:responsiblePerson;
622     rr:objectMap [ rr:column "[CeleJmeno]" ];
623 ];
624
625 <#AmendmentVersion> a rr:TriplesMap;
626     rr:logicalTable <#AmendmentTableView>;
627     rr:subjectMap [
628         rr:template "http://localhost:7598/amendment/{ID}/{PoradiVerzeDodatku}/
version";
629         rr:class cn:Version;
630     ];
631     rr:predicateObjectMap [
632         rr:predicate cn:uri;
633         rr:objectMap [ rr:template "http://localhost:7598/amendment/{ID}/{
PoradiVerzeDodatku}" ];
634     ];
635     rr:predicateObjectMap [
636         rr:predicate cn:versionOrder;
637         rr:objectMap [
638             rr:column "[PoradiVerzeDodatku]";
639             rr:datatype xsd:integer;
640         ];
641     ];
642     rr:predicateObjectMap [
643         rr:predicate dc:issued;
644         rr:objectMap [
645             rr:column "[DATUMZMENYSTAVUJIS]";
646             rr:datatype xsd:dateTime;
647         ];
648     ];
649
650 <#Attachment> a rr:TriplesMap;
651     rr:logicalTable <#AttachmentTableView>;
652     rr:subjectMap [
653         rr:template "http://localhost:7598/attachment/{ID}/1";
654         rr:class cn:Attachment;
655     ];
656     rr:predicateObjectMap [
657         rr:predicate dcmi:type;
658         rr:objectMap [ rr:constant "Příloha" ];
659     ];
660     rr:predicateObjectMap [
661         rr:predicate cn:anonymised;
662         rr:objectMap [
663             rr:column "[Anonymizovano]";
664             rr:datatype xsd:boolean;
665         ];
666     ];
667     rr:predicateObjectMap [
668         rr:predicate dc:title;
669         rr:objectMap [ rr:column "[POPIS_NAZEV]" ];
670     ];
671     rr:predicateObjectMap [
672         rr:predicate dc:identifier;
673         rr:objectMap [ rr:column "[ID]" ];
674     ];
675     rr:predicateObjectMap [
676         rr:predicate cn:valid;
677         rr:objectMap [
678             rr:template "true";
679             rr:datatype xsd:boolean;
680         ];
681     ];
682     rr:predicateObjectMap [
683         rr:predicate schema:url;
684         rr:objectMap [ rr:template "http://localhost:7598/file/{
SADADUL_ULOSTEID}/{NAZEVSOUBORU}" ];
685     ];
686     rr:predicateObjectMap [
687         rr:predicate dc:publisher;
688         rr:objectMap [ rr:template "http://localhost:7598/publisher" ];

```

```

689 ];
690 rr:predicateObjectMap [
691   rr:predicate cn:version;
692   rr:objectMap [ rr:template "http://localhost:7598/attachment/{ID}/1/
version" ];
693 ];
694
695 <#AttachmentToContract> a rr:TriplesMap;
696   rr:logicalTable <#AttachmentToContractTableView>;
697   rr:subjectMap [
698     rr:template "http://localhost:7598/attachment/{ID}/1"
699   ];
700   rr:predicateObjectMap [
701     rr:predicate cn:contract;
702     rr:objectMap [ rr:template "http://localhost:7598/contract/{SmlouvaID}/{
PORADIVERZE}" ];
703 ];
704
705 <#AttachmentVersion> a rr:TriplesMap;
706   rr:logicalTable <#AttachmentTableView>;
707   rr:subjectMap [
708     rr:template "http://localhost:7598/attachment/{ID}/1/version";
709   rr:class cn:Version;
710 ];
711   rr:predicateObjectMap [
712     rr:predicate cn:uri;
713     rr:objectMap [ rr:template "http://localhost:7598/attachment/{ID}/1" ];
714 ];
715   rr:predicateObjectMap [
716     rr:predicate cn:versionOrder;
717     rr:objectMap [
718       rr:template "1";
719       rr:datatype xsd:integer;
720     ];
721 ];
722   rr:predicateObjectMap [
723     rr:predicate dc:issued;
724     rr:objectMap [
725       rr:column "[OKAMZIKVYTVORENI]";
726       rr:datatype xsd:dateTime;
727     ];
728 ];
729
730 <#ContractToImplementations> a rr:TriplesMap;
731   rr:logicalTable <#MilestoneTableView>;
732   rr:subjectMap [
733     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}";
734   ];
735   rr:predicateObjectMap [
736     rr:predicate cn:implementation;
737     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}/implementation" ];
738 ];
739
740 <#Implementation> a rr:TriplesMap;
741   rr:logicalTable <#MilestoneTableView>;
742   rr:subjectMap [
743     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}/
implementation";
744   rr:class cn:Implementation;
745 ];
746   rr:predicateObjectMap [
747     rr:predicate cn:milestone;
748     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}/milestone/{MilestoneID}" ];
749 ];
750
751 <#Milestone> a rr:TriplesMap;
752   rr:logicalTable <#MilestoneTableView>;
753   rr:subjectMap [
754     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}/milestone
/{MilestoneID}";
755   rr:class cn:Milestone;

```

```
756 ];  
757 rr:predicateObjectMap [  
758   rr:predicate dc:title;  
759   rr:objectMap [ rr:column "[NAZEV]" ];  
760 ];  
761 rr:predicateObjectMap [  
762   rr:predicate cn:dueDate;  
763   rr:objectMap [  
764     rr:column "[DATUMUCINOSTIML]";  
765     rr:datatype xsd:dateTime;  
766   ];  
767 ].
```

R2RML skript mapující smlouvy