Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky

Přepoužívání ontologických modelů na sémantickém webu

Teze profesorské přednášky

Autor: Doc. Ing. Vojtěch Svátek, Dr.
Obor jmenovacího řízení: Aplikovaná informatika

Praha, duben 2017

Předmluva

Tento text představuje teze přednášky konané v rámci řízení ke jmenování profesorem před Vědeckou radou Fakulty informatiky a statistiky Vysoké školy ekonomické v Praze dne 4.5.2017. Přednáška i předložený text mají za cíl seznámit s problematikou sémantického webu, rolí ontologických modelů, a motivacemi pro jejich přepoužívání. Podrobněji se zaměří na soubor metod vyvinutých autorem a jeho týmem, které mají za cíl usnadnit a zkvalitnit proces přepoužívání ontologických modelů. Prezentované výsledky byly publikovány v impaktovaných časopisech, zejména *J. Web Semantics*, ve sbornících předních evropských a světových konferencí, např. *Intl' Conf. on Knowledge Engineering and Knowledge Management* (EKAW), *Extended Semantic Web Conference* (ESWC), *Intl' Conf. on Knowledge Capture* (K-CAP) nebo *Intl' Conf. on Formal Ontology in Information Systems* (FOIS), a formou kapitol v zahraničních editovaných knihách.

Autor by rád na tomto místě vyjádřil velké poděkování spolupracovníkům, kteří významným podílem přispěli k dosažení výsledků popsaných v tomto textu (podrobnější rozbor týmové spolupráce je uveden v sekci 5), dále pak vedoucímu katedry V. Sklenákovi a všem dalším kolegům, kteří jeho vědecké aktivity v rámci katedry podporovali.

Výzkum byl spolufinancován z grantových prostředků následujících projektů (souvislost mezi výzkumnými výsledky a projekty je rovněž zachycena v sekci 5): projekty EU Knowledge Web (IST FP6-507482) a LOD2 (ICT FP7-257943), projekt GA ČR P202/10/1825 (PatOMat), mobilitní projekt MŠMT 7AMB12SK020, a projekty IGA VŠE 12/06, 20/07, F4/34/2014, F4/90/2015 a F4/28/2016.

1 Úvod

Pojem *ontologie* se od 17. století používal v dílech filozofických autorů, a to zejména jako označení pro disciplínu zabývající se "jsoucny" (tj. "věcmi", které v určitém smyslu "jsou"), bytím jako takovým, a základními filozofickými pojmy. V informatice se od 90. let 20. století začal tento pojem používat v posunutém významu: pro formalizovaný popis (model) určité problematiky, tj. zachycení toho, co v dané tematické oblasti "existuje". Asi nejznámější definicí ontologie z tohoto období je Gruberova [10] "explicitní specifikace konceptualizace" z r. 1993. Vstupními pojmy definice jsou "konceptualizace", představující systém provázaných pojmů vzniklý (zpravidla) v lidské mysli, a "explicitní specifikace", chápaná jako vyjádření v určitém sdíleném jazyce – zpravidla formálně-logickém, i když za ontologie jsou někdy považovány i polostrukturované glosáře s pojmy definovanými v přirozeném jazyce. Výzkum, popřípadě praktické aplikace, v oblasti informatických ontologií byly během 90. let záležitostí úzkého okruhu specialistů, převážně z oblasti umělé inteligence. Hlavním způsobem využívání ontologií bylo formálně-logické odvozování nových faktů z již známých, popřípadě kontrola logické konzistence celé soustavy pojmů. Ontologie byly v té době rozsáhlé, propracované a zpravidla monolitické. Jejich tvorba a další zpracování se staly předmětem nově vzniklého podoboru znalostního inženýrství označovaného jako *ontologické inženýrství*.

Na samém konci milénia dostalo ontologické inženýrství nový impuls, pocházející z oblasti rovněž nově vznikajících *webových technologií*. Web se v té době stále více stával už ne jen médiem pro sdílení propojených textových dokumentů mezi lidmi, ale i potenciální platformou pro spolupráci softwarových aplikací. Aplikace vyžadují data ve *strukturované* podobě, a pokud možno popsaná *schématy*, která jsou rovněž strojově zpracovatelná, aby sebemenší změna používaného datového schématu (která je v otevřeném webovém prostředí velmi běžná) nevyžadovala zásah programátora. Idea využít pro konstrukci webových datových schémat již částečně prověřený koncept ontologických modelů[1] byla velmi přirozená, a

[1] V textu termín "ontologie" a "ontologický model" dále používáme víceméně záměnně (s tím, že přinejmenším pro modely ontologického pozadí popisované v sekci 4.3, případně i pro sady tzv. mapovacích tvrzení

ontologie byly proto od první chvíle chápány jako ústřední prvek tzv. *sémantického webu*: proklamované "nové generace" webu, kde budou mít informace/data přiřazena strojově zpracovatelný význam (sémantiku).

Na rozdíl od nejranější éry ontologického inženýrství v tomto období (a to až do současnosti) narůstá význam *rozsáhlých datových zdrojů*, které primárně neslouží pro odvozování zcela nových informací (jako tomu bylo u znalostních bází systémů umělé inteligence), ale spíše je potřeba je efektivně třídit a vyhledávat v nich. Namísto snahy vyčerpávajícím způsobem pokrýt všechny pojmy z určité věcné oblasti je výběr zahrnutých pojmů zahrnutých do určité ontologie ovlivňován pozorováním, jaké typy objektů nejčastěji jsou či mají být v datech vystavovaných ("publikovaných") na webu popisovány. Cílem je dosáhnout jejich *přepoužitelnosti* pro co největší počet datových sad, jednak proto, aby se jednorázově vložené úsilí do návrhu ontologie co nejvíce zúročilo, ale zejména proto, že datové sady popsané stejnými ontologickými entitami je možné dobře slučovat nebo se nad nimi federovaně dotazovat, a mohou si je snadno předávat jednotlivé webové aplikace, které se tak stávají lépe *interoperabilními*.

Jak ukazují nedávné studie, např. Schaibleho [20], přepoužívání ontologií pro nově vznikající datové sady v praxi naráží na řadu překážek. Některé z nich mají převážně *socio-ekonomický* charakter, např. to, že velké firmy i v této oblasti prosazují využívání svých standardů, byť ne vždy ideálně navržených; za asi nejvýznamnější překážkou *technologického* charakteru však lze považovat *strukturní heterogenitu* ontologií, které jsou jinak, z věcného hlediska, zcela relevantní pro přepoužití. Pokud totiž ontologie modeluje stejnou realitu pomocí jiných – nebo jinak propojených – konstruktů formálního jazyka, než odpovídá struktuře nově publikovaných dat, nelze na ni z těchto dat přímočaře odkazovat, a datové sady, ač vystavené na webu, tak zůstávají sémanticky nepropojenými "ostrovy".

Technologické[2] aspekty přepoužívání ontologií (na sémantickém webu), se zvláštním ohledem na problém strukturní heterogenity, tvoří významnou část výzkumných prací, kterými se autor (spolu s týmovými kolegy) v průběhu posledních přibližně 12 let zabýval. Problém strukturní heterogenity ontologií sémantického webu zřejmě poprvé zformuloval v r. 2007 F. Scharffe v úzkém kontextu tzv. *mapování ontologií* (které je jen jedním z mnoha scénářů přepoužití ontologií). Autor studium tohoto problému, ve spolupráci se svým doktorandem O. Švábem a právě F. Scharffem (INRIA, Francie) v r. 2009 rozšířil na širší spektrum úloh ontologického inženýrství (zejména na volbu strukturního "stylu" při tvorbě nové ontologie), a také poukázal na možnost řešit ho nejen pomocí mapování ontologií, ale i pomocí jejich *transformace*. Novým přínosem autora v tomto směru byl mj. i rozbor možností *lexikálních* vzorů pro transformaci ontologií. V reakci na úskalí spojená s párovou transformací mezi strukturními styly (kombinatorická složitost a obtížná správa vzorů) autor následně, v r. 2012, zformuloval zcela nový koncept *modelu ontologického pozadí* jako "mediačního" prostředku odlišných stylů, a ve spolupráci se spolupracovníkem M. Vacurou a kolegy z UK Bratislava pro tyto modely vyvinul reprezentační jazyk nazvaný PURO. Jedním z jeho hlavních využití je přepoužívání společných pojmů v rámci variantních ontologií založených na odlišných strukturních stylech. Zatím posledním zásadnějším "přírůstkem" do rodiny metod řešících problém strukturní heterogenity v kontextu přepoužívání ontologií pak je metoda výpočtu tzv. *fokusované kategorizační síly* ontologií, která umožňuje prostřednictvím zvolené sady výrazů ontologického jazyka zohlednit odlišné strukturní styly ontologií při rozhodování o jejich přepoužití.

V souhrnu lze říci, že soubor modelů a metod vyvinutých autorem a jeho týmem postupně získal v *mezinárodním výzkumu* ontologického inženýrství výrazný ohlas, reflektovaný i řadou

napříč ontologiemi, je vhodnější používat obecnější termín "ontologický model"). Téměř shodný význam má i termín "datový slovník".

[2] V rámci projektů zaměřených na propojená data (zejména projekty EU LOD2 nebo OpenBudgets.eu) se autor do jisté míry věnoval i socio-ekonomickým aspektům přepoužívání datových slovníků, tato tématika však již leží mimo záběr profesorské přednášky.

společných publikací se zahraničními špičkami a pozváním do projektů rámcových programů EU. Pokud jde o přímé nasazení *v praxi*, došlo k němu zatím jen ve velmi omezené míře (v rámci rozsáhlejších nástrojů vyvinutých zahraničními partnery), což ovšem souvisí spíše s určitou "rezervovaností" praxe vůči netriviálním ontologickým modelům po opadnutí počáteční "módní vlny" sémantického webu.[3] V současnosti si praxe potřebu ontologické sémantiky datových modelů začíná opět více uvědomovat (viz např. začleňování propracovaných oborových extenzí do ontologického modelu *schema.org* podporovaného Googlem a dalšími velkými firmami), a lze očekávat, že vhodná prototypová řešení pro zpracování netriviálních ontologií budou v podnikovém a veřejnosprávním prostředí opět stále více nasazována.

Zbylá část textu je strukturována následovně:

- Sekce 2 nastiňuje roli hlavních *reprezentačních jazyků* používaných na sémantickém webu, RDF a OWL, a představuje na příkladech charakteristické typy ontologických modelů používaných v tomto prostředí.
- Sekce 3 uvádí čtenáře do vybraných základních pojmů ontologického inženýrství se specifickým přihlédnutím k těm, které jsou klíčové pro dále popisovaný vlastní výzkum: jedná se o *přepoužití* ontologií, *mapování* ontologií (jako jeho specifickou variantu), ontologické (návrhové a empirické) *vzory*, a *strukturní heterogenitu* ontologií. V rámci rozboru strukturní heterogenity je zvláště zmíněna ontologická kolekce OntoFarm, specificky vyvinutá autorovým týmem pro potřebu studia tohoto fenomenu.
- Sekce 4 se postupně věnuje čtyřem odlišným *scénářům* úlohy přepoužití ontologie a pro ně vyvinutým metodám:
  - přepoužití ontologie s využitím komplexních *mapovacích vzorů* (Sekce 4.1);
  - přepoužití ontologie "nejlepší praxe", popřípadě návrhového vzoru, do jiné ontologie, zahrnující *strukturní transformaci* přijímající ontologie (Sekce 4.2); stručně jsou přitom zmíněny i další, související případy užití transformace založené na transformačních vzorech;
  - využití tzv. *modelů ontologického pozadí*, které umožňují jak zachytit pokrytí stejných pojmů různými existujícími ontologiemi v odlišné struktuře, tak i vytvářet ontologie nové v různých strukturních variantách stejného věcného obsahu (Sekce 4.3);
  - přepoužití ontologie s ohledem na její (fokusovanou) *kategorizační sílu* (Sekce 4.4).
- Předposlední sekce 5 uvádí popisovaný výzkum do kontextu vědecké činnosti katedry jako celku i v jejím rámci působící pracovní skupiny "Semantic Web and Ontological Engineering" (SWOE). Je zde vymezen podíl klíčových spolupracovníků na výzkumu i podpora ze strany grantových projektů.
- Závěrečná sekce 6 pak již jen velmi stručně shrnuje obsah tezí.

Publikovaných (v drtivé většině zahraničních) prací autora na tématech spadajících pod zaměření profesorské přednášky lze celkově dohledat přes 60. V seznamu referencí uvedeném na konci dokumentu je z úsporných důvodů zařazena jen jejich menší část (zpravidla ty práce, kde je daný problém nebo metoda představen/a v nejrozpracovanější podobě). Seznam všech publikovaných prací autora lze dohledat v publikační databázi PCVSE, `http://pcvse.vse.cz`, kde je možné je i filtrovat podle čísel grantových projektů.

[3] K tomuto opadnutí došlo celosvětově zhruba v letech 2004–2006, tj. právě v době, kdy se výzkum autora v oboru začal plně rozvíjet. Jeho hlavní příčinou byl tehdejší nadměrný důraz části akademické sféry na využívání komplexních (uživatelsky neintuitivních a výpočetně náročných) postupů formálně-logického odvozování v aplikacích, které ho reálně nevyžadovaly. Nedůvěra k "ontologickým" řešením pak paradoxně vedla i k odmítání přístupů, které naopak usilovaly o zajištění lepší srozumitelnosti dat a zvýšení efektivity jejich zpracování. Podle názoru autora většina výzkumu jeho týmu spadá do této druhé kategorie.

2 Reprezentace dat a znalostí na sémantickém webu

V této části se nejprve budeme věnovat reprezentačním jazykům sémantického webu, o které se "hlavní proud" soudobého ontologického inženýrství opírá jako o svůj syntaktický a formálně logický základ, a ukážeme si na příkladech charakteristické typy používaných ontologických modelů.

2.1 RDF – Resource Description Framework

Základním reprezentačním jazykem sémantického webu je *RDF* (Resource Description Framework) [1]. Data reprezentovaná v jazyce RDF mají podobu trojic "subjekt – predikát – objekt", kde v roli subjektu a predikátu vždy vystupují globálně[4] unikátními identifikátory *IRI*[5] (strukturovanými podle webového protokolu HTTP, tj. analogicky k adresám URL běžných webových dokumentů); objekt může být vyjádřen jako IRI nebo přímo jako datová hodnota, tzv. *literál*, odpovídající určitému datovému typu (např. řetězec, celé číslo nebo datum). Každá trojice[6] by měla být interpretovatelná jako tvrzení, kterým je subjektu přiřazen objekt jako hodnota predikátu. Trojice můžeme dále rozdělit na tři typy:

- *Instanciace*, `<IRI1> rdf:type <IRI2>`, který interpretujeme tak, že subjekt `<IRI1>` je instancí objektu (v tomto případě tedy *třídy*) `<IRI2>`
- *Vztah* (jiný než instanciace), `<IRI1> <IRI2> <IRI3>`, který interpretujeme tak, že subjekt `<IRI1>` je ve vztahu `<IRI2>` k objektu `<IRI3>`
- *Přiřazení dat*, `<IRI1> <IRI2> <Literal>`, který interpretujeme tak, že predikát `<IRI2>` nabývá pro subjekt `<IRI1>` datové hodnoty `<Literal>`.

Instanciace je tedy definována pomocí vyčleněného predikátu, který zde máme uvedený stručným zápisem v prefixovaném tvaru: `rdf:type`. Ten je standardním zkráceným zápisem plného IRI `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`, s tím, že přiřazení prefixu `rdf` jmennému prostoru `http://www.w3.org/1999/02/22-rdf-syntax-ns#` (jmenný prostor datového slovníku jazyka RDF samotného[7]) bývá uvedeno v hlavičce souboru s daty RDF (v těch syntaktických variantách zápisu dat RDF, které takové zkracování umožňují). Podobným způsobem, tj. pomocí prefixovaných jmen, kdy prefix odpovídá určitému jmennému prostoru (vyjadřující příslušnost k datovému slovníku nebo datové sadě) se zapisují i jiné predikáty, ale i subjekty a objekty. Příklady trojic výše uvedených tří typů tedy mohou být např. (`ex` je prefix používaný pro ukázková data v učebnicových příkladech):

- *Instanciace*: `ex:Praha rdf:type ex:Mesto`
- *Jiný vztah*: `ex:Praha ex:zeme ex:CeskaRepublika`
- *Přiřazení dat:* `ex:Praha ex:pocetObyvatel 1200000`

[4] Pro přesnost, standard RDF povoluje i lokální identifikátory platné jen v rámci určitého datového souboru, tzv. "blank nodes". Jejich používání však autority v případě dat určených pro veřejné vystavování, vesměs nedoporučují a lze se mu vyhnout.

[5] International Resource Identifier; jde o rozšířenou variantu identifikátoru URI, Universal Resource Identifier, oproti kterému navíc umožňuje používat znaky z ne-ASCII abeced.

[6] V současnosti se stále více používá model RDF založený na čtveřicích. Čtvrtým prvkem pak je identifikátor tzv. pojmenovaného grafu: datové množiny, do které daná trojice patří. Tímto způsobem lze data jednoduše modularizovat. Pro účely tohoto textu však pojmenované grafy není nutno uvažovat, protože na úlohu přepoužívání ontologií nemají významný vliv.

[7] Jazyk RDF totiž definuje svůj vlastní metamodel stejným způsobem, jako jsou v něm definovány různé specifičtější datové slovníky (ontologie).

Již nyní se zastavme u otázky tzv. *množinové interpretace* některých prvků dat RDF, kterou lze na těchto příkladech dobře vysvětlit. Je zjevné, že `ex:Mesto` je třídou, kterou lze interpretovat jako *množinu jejích instancí*, tj. konkrétních měst (`ex:Praha`, `ex:Brno`, `ex:Ostrava`, atd.). Obdobně lze ale i predikát `ex:zeme` interpretovat jako množinu *uspořádaných dvojic* (<*město*>,<*země*>), tj., matematicky, jako binární relaci, tj. podmnožinu kartézského součinu množiny měst a množiny zemí. Na rozdíl od toho, interpretací IRI `ex:Praha` nebo `ex:CeskaRepublika` je pouze jednotlivý objekt,[8] *instance*, neboli také *individuál*[9] – termín "instance" totiž evokuje existenci nějaké třídy, do které by objekt měl patřit; v otevřeném prostředí sémantického webu však (na rozdíl od různých "uzavřených" objektově orientovaných datových systémů) nemusí být individuální objekty od svého vzniku navázané na třídy.

Součástí metamodelu jazyka RDF jsou i prostředky pro tvorbu datových slovníků, sdružené ve jmenném prostoru `http://www.w3.org/2000/01/rdf-schema#` (RDF Schema, standardní prefix `rdfs`). Definice slovníků mají opět podobu množin trojic RDF, ve kterých se jako predikáty používají tyto čtyři:[10] `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain` a `rdfs:range`:

- `rdfs:subClassOf`, vyjadřující, že třída v subjektu trojice s tímto predikátem je *podtřídou* třídy v objektu dané trojice;
- `rdfs:subPropertyOf`, vyjadřující, že *vlastnost* (binární relace) v subjektu trojice s tímto predikátem je *podvlastností* (tj. podrelací) vlastnosti v objektu dané trojice;
- `rdfs:domain`, vyjadřující, že *definičním oborem* vlastnosti v subjektu trojice s tímto predikátem je množina všech instancí třídy v objektu dané trojice;
- `rdfs:range`, vyjadřující, že *oborem hodnot* vlastnosti v subjektu trojice s tímto predikátem je množina všech instancí třídy v objektu dané trojice.

Vidíme, že na úrovni schématu již odlišujeme třídy a vlastnosti (binární relace mezi prvky tříd). V "instančních" datech odpovídajících danému schématu se pak IRI vlastností přímo využívají jako predikáty, zatímco objekty a subjekty v trojicích z "instančních" dat se na schémata odkazují pomocí instanciace (s využitím predikátu `rdf:type`).
Pozn.: Rozdíl mezi pojmy "predikát" a "vlastnost" je ten, že predikát je zpravidla chápán určen pozičně (jako druhý prvek v rámci trojic/e RDF), zatímco vlastnost je vše, co *může* být v roli predikátu použito; proto se mohou ve "schématových" trojicích vlastnosti vyskytovat i v pozici subjektu a/nebo objektu, např. můžeme říci, že vlastnost `ex:jeHlavnimMestem` je podvlastností vlastnosti `rdfs:nachaziSeV`:

```
ex:jeHlavnimMestem rdf:subPropertyOf ex:nachaziSeV
```

Tuto trojici množinově interpretujeme tak, že množina uspořádaných dvojic *hlavních měst* a jejich zemí je podmnožinou množiny uspořádaných dvojic všech lokalit a jejich zemí.

Jazyk RDF (resp. RDF Schema) již umožňuje jednoduché logické odvozování, např. z existence trojic

```
ex:Praha rdf:type ex:Mesto
ex:Mesto rdfs:subClassOf ex:Lokalita
```

jednoduchým způsobem odvodíme novou trojici

[8] Zde mluvíme o "objektu" v obecném smyslu, nikoliv ve smyslu pozice určitého IRI v rámci trojice RDF.
[9] Tento výraz se běžně používá v deskripční logice, o které bude řeč dále.
[10] Pokud se omezíme na predikáty s formálně logickou interpretací. Součástí RDF Schema jsou ovšem i další, podpůrné predikáty. Tak např. `rdfs:label` umožňuje entitě `ex:Praha` přiřadit hodnotu (řetězcový literál) "Praha".

```
ex:Praha rdf:type ex:Lokalita
```

Obdobně funguje odvozování nad uspořádanými dvojicemi v případě `rdfs:subPropertyOf`. Z trojic

```
ex:Praha ex:jeHlavnimMestem ex:CeskaRepublika
ex:jeHlavnimMestem rdfs:subPropertyOf ex:nachaziSeV
```

odvodíme

```
ex:Praha ex:nachaziSeV ex:CeskaRepublika
```

O trochu méně intuitivní, ale neméně logicky korektní, je využití `rdfs:domain`; z trojic

```
ex:Praha ex:jeHlavnimMestem ex:CeskaRepublika
ex:jeHlavnimMestem rdfs:domain ex:Mesto
```

můžeme odvodit

```
ex:Praha rdf:type ex:Mesto
```

Predikát `rdfs:range` funguje analogicky. Z trojic

```
ex:Praha ex:jeHlavnimMestem ex:CeskaRepublika
ex:jeHlavnimMestem rdfs:range ex:Zeme
```

odvodíme

```
ex:CeskaRepublika rdf:type ex:Zeme
```

K tvorbě datových slovníků a schémat ještě poznamenejme, že zpravidla probíhá odděleně od tvorby a zpracování "instančních" dat. V praxi bychom proto pro obecné třídy a vlastnosti, jako je `jeHlavnimMestem`, `nachaziSeV`, `Mesto` nebo `Zeme` používali odlišný jmenný prostor než pro individuály jako `Praha` nebo `CeskaRepublika`. I z hlediska způsobu vystavování a zpřístupňování je mezi sadami "instančních" a "schematových" dat zpravidla rozdíl, jehož vysvětlení přesahuje záběr tohoto textu. Pro problematiku technologie tvorby a vystavování dat (instancí i schémat) na sémantickém webu je dobrým zdrojem např. kniha *Linked Data: Evolving the Web into a Global Data Space* [12].

2.2 OWL – Web Ontology Language

Jazyk *OWL* (Web Ontology Language) [2] je výrazným rozšířením RDF Schema (který je, z hlediska používaných výrazových prostředků, jeho podmnožinou), podporující rozsáhlé odvozování nad vytvořeným datovým schématem. K tomu mu slouží rigorózní mapování na tzv. deskripční logiku [3]. Z vyjadřovacích prostředků OWL vybereme ty, na které se budeme později odvolávat při vysvětlování nově navržených metod.

Relativně drobným rozšířením oproti RDF Schema je zavedením predikátu pro ekvivalenci tříd, `owl:equivalentClass`. Pomocí něj můžeme vyjádřit, že dvěma třídám odpovídá shodná množina instancí, např.

```
ex:Zeme    owl:equivalentClass  ex:Stat
```

(Stejného efektu lze dosáhnout i pomocí RDF Schema, tím, že každou ze tříd označíme jako podtřídu té druhé, je to však podstatně méně elegantní.)

Zřejmě nejmarkantnějším rozšířením oproti RDF Schema je zavedení tzv. *složených konceptových výrazů*, se kterými lze pracovat stejně jako s běžnými, tzv. *pojmenovanými* třídami, avšak namísto jednoho IRI se na ně odvoláme pomocí kombinace IRI a konstruktů OWL.

Jedním ze nejvyužívanějších složených výrazů v OWL je tzv. *existenční restrikce*, což je konceptový výraz s obecným schématem, v notaci deskripční logiky, $\exists R.C$, kde R je vlastnost a C je "cílová" třída. Množinovou interpretací tohoto výrazu je "množina individuálů, které jsou spojeny vlastností R s (alespoň jednou) instancí třídy C". Tímto způsobem můžeme vymezit např. třídu všech hlavních měst, tj. entit spojených vlastností `ex:capitalOf` s nějakou instancí třídy `ex:Zeme`, a přiřadit takovou "anonymní" třídu (složený konceptový výraz) určité instanci. V notaci deskripční logiky[11] může příslušné *tvrzení* vypadat takto:

$Praha \in \exists\ jeHlavnimMestem\ .\ Zeme$

Poznamenejme, že přestože je ovšem takové tvrzení stále jednoduché, již ho nelze popsat jedinou trojicí RDF, ale potřebujeme jich několik:

```
ex:Praha    rdf:type           _:a
_:a         rdf:type           owl:Restriction
_:a         owl:onProperty     ex:jeHlavnimMestem
_:a         owl:someValuesFrom ex:Zeme
```

Entita `_:a` je zde zástupným symbolem za celý složený konceptový výraz (instanci třídy konceptových výrazů `owl:Restriction`), a odkazuje se specializovanými schématovými predikáty `owl:onProperty` a `owl:someValuesFrom` na své složky.

Vraťme se nyní k otázce logického odvozování. Ve slovníku (ontologii) můžeme např. nadefinovat vztah ekvivalence mezi pojmenovanou třídou `ex:HlavniMestoZeme` a výše uvedeným konceptovým výrazem jako

```
ex:HlavniMestoZeme   owl:equivalentClass  _:a
_:a                  rdf:type             owl:Restriction
_:a                  owl:onProperty       ex:jeHlavnimMestem
_:a                  owl:someValuesFrom   ex:Zeme
```

neboli, v notaci deskripční logiky, jako

$HlavniMestoZeme \equiv \exists\ jeHlavnimMestem\ .\ Zeme$

S pomocí takového obecného vztahu pak můžeme z trojic

```
ex:Praha ex:jeHlavnimMestem ex:CeskaRepublika
ex:CeskaRepublika rdf:type ex:Zeme
```

odvodit

```
ex:Praha rdf:type ex:HlavniMestoZeme
```

Na první pohled se takové odvození zdá podobné tomu, které jsme realizovali pomocí `rdfs:range`, když jsme stejnému individuálu přiřadili třídu `ex:Mesto`. Odlišnost ale tkví v přítomnosti cílové třídy v restrikci. Pokud totiž budou vstupními trojicemi např.

```
ex:Ostrava ex:jeHlavnimMestem ex:MoravskoslezskyKraj
ex:MoravskoslezskyKraj rdf:type ex:Kraj
```

chybná trojice

```
ex:Ostrava rdf:type ex:HlavniMestoZeme
```

se neodvodí, zatímco v případě použití `rdfs:range` by k jejímu odvození došlo.

[11] V matematické notaci deskripční logiky zpravidla používáme jen jednoduchá jména entit, a nikoliv jmenné prostory, pokud to není nutné.

2.3 Webové ontologie a jejich typy

Ontologie je v kontextu sémantického webu zpravidla chápána (ve formálně-logickém smyslu) jako množina schématových tvrzení jazyka OWL. Množina všech entit (tříd, vlastností, popřípadě významných individuálů) vyskytujících se v těchto tvrzeních je označována jako *signatura* dané ontologie. Z praktického hlediska lze dále jako součást ontologie chápat např. i její lidmi napsanou slovní dokumentaci, usnadňující její použití.

V rámci rozsáhlé množiny populárních ontologií lze identifikovat několik (neostře ohraničených) typů, jejichž struktura je ovlivněná způsobem jejich využívání. Jedná se o ontologie pro automatické klasifikování, značkovací slovníky, a slovníky pro publikování propojených dat (linked data). Způsob využívání má vliv i na strukturu ontologie.

Ontologie pro automatické klasifikování se používají převážně v oblasti biomedicíny a zakládají se na zdravotnických klasifikačních systémech vyvíjených po desítky let. Tyto ontologie typicky nejsou primárně využívány spolu s "instančními" datovými sadami; pozornost se soustřeďuje na *třídy*, představující abstraktní prototypy reálných "věcí" (např. částí těla, chorob a jejich původců, lékařských zákroků apod.) Automatickým klasifikováním se myslí vytvoření explicitní vazeb mezi pojmenovanými třídami pomocí logického odvozovacího nástroje, na základě modulárních definic těchto tříd (velmi často využívajících již zmíněnou existenční restrikci, případně další formy restrikcí, zejména tzv. univerzální nebo kardinalitní restrikci). Příkladem takového odvození může být automatické podřazení pojmu "vynětí slepého střeva" obecnějšímu pojmu "operace střev", tj. vytvoření nové schematové trojice $Appendicectomy \sqsubseteq IntestinalSurgery$, pokud v ontologii[12] platí množina výchozích tvrzení

$Appendicectomy \equiv SurgicalProcedure \sqcap \exists\ method.Excision \sqcap \exists\ site.AppendixStructure$
$IntestinalSurgery \equiv SurgicalProcedure \sqcap \exists\ site.IntestinalStructure$
$AppendixStructure \sqsubseteq IntestinalStructure$

Pozn.: symbol $\sqsubseteq$ v notaci deskripční logiky označuje vztah podtřídy (`rdfs:subClassOf`) a symbol $\sqcap$ logickou konjunkci.

Značkovací slovníky se rozvinuly v rámci snahy *vyhledávacích portálů* na jedné straně zpřehlednit výsledky vyhledávání pro uživatele a na druhé straně posbírat z indexovaných stránek strukturovaná data využitelná jak technologicky (pro zdokonalování vyhledávačů), tak komerčně. Značkovací slovníky jsou primárně určeny pro anotování různorodých webových stránek jednoduše vytvořitelnými metadaty a typicky se skládají, pro danou věcnou doménu, z několika málo tříd a naopak poměrně velkého počtu *vlastností*, jejichž obor je velmi často *literálový*[13] (tj. textový řetězec, číslo, nebo např. URL). Toto řešení je zvoleno právě kvůli požadavku na jednoduchost pro webové návrháře. Nejrozšířenějším značkovacím slovníkem je v současnosti *schema.org*, jehož vznik iniciovaly právě velké vyhledávače (Google, Bing a Yahoo!). Jako konkrétní příklad si z něj můžeme uvést třídu `http://schema.org/LocalBusiness`, definující fyzickou provozovnu podniku poskytujícího prodej nebo služby. Pro tuto třídu jsou doporučeny čtyři specifické vlastnosti (vedle těch "zděděných" od jejích nadtříd jako je `schema:Organization`):[14] `schema:currenciesAccepted`, `schema:openingHours`, `schema:paymentAccepted` a `schema:priceRange`, přičemž ve všech případech je jejich oborem hodnot `schema:Text`, tedy textový řetězec. Jednoznačnosti hodnot takových vlastností je dosahováno nepřímo, pomocí odkazů na textové kódovníky (např.

[12] Příklad byl lehce upraven z ontologie SNOMED-OWL, viz `https://bioportal.bioontology.org/ontologies/SNOMEDCT`. Názvy entit byly ponechány v angličtině kvůli riziku nepřesného překladu anglických medicínských termínů.

[13] Takovým vlastnostem v OWL říkáme *datové*, na rozdíl od *objektových* vlastností, jejichž obor je vymezen některou ontologickou třídou.

[14] Dále už pro úspornost uvádíme jména entit s prefixem `schema` zastupujícím `http://schema.org/`.

ISO 4217 pro měny) nebo instrukcemi pro doporučený tvar rozsáhlejšího textu (např. v případě otvírací doby je doporučeno označovat dny v týdnu dvouznakovými zkratkami a oddělovat je od sebe čárkami). Strojová srozumitelnost těchto dat je omezená a vyžaduje zpracování prostředky textového inženýrství, např. pomocí regulárních výrazů.

Slovníky pro propojená data jsou vyvíjeny v návaznosti na iniciativu *LinkedData.org*, iniciovanou autorem koncepce WWW, Timem Berners-Lee. Iniciativa usiluje o to, aby strukturovaná data byla nejen vystavována na webu, ale také propojována odkazy mezi jednotlivými entitami napříč nezávisle vzniklými datovými sadami. Repozitář slovníků spadajících (převážně) do této kategorie, *Linked Open Vocabularies*[15] (LOV), aktuálně obsahuje přibližně 600 modelů z různých oblastí. Na rozdíl od značkovacích slovníků je zde kladen důraz na využití *objektových* vlastností, jejichž hodnotami jsou IRI. Tímto způsobem je možné přímo odkazovat z jedné datové sady do jiné. Známým příkladem slovníku pro propojená data je FOAF (Friend of a Friend),[16] slovník určený pro zachycení informací o lidech a jejich vzájemných kontaktech, tj. vytváření otevřené, sémantické sociální sítě. Většina vlastností z tohoto slovníku jsou objektové, jako např. `foaf:homepage`, přiřazující něčemu nebo někomu domovskou stránku (jako instanci třídy `foaf:Document`, která může být sama popsána dalšími vlastnostmi) nebo `foaf:knows`, přiřazující člověku jiného člověka (v obou případech jde o instanci třídy `foaf:Person`), se kterým se zná.

Dalšími velmi rozšířenými slovníky pro propojená data jsou *Simple Knowledge Organization System*[17] (SKOS) a *Data Cube Vocabulary*[18] (DCV), které na rozdíl od FOAF nemodelují realitu samotnou, ale slouží jako reprezentační prostředek pro systémy strukturování znalostí: v případě SKOS hierarchicky (např. pro potřebu tvorby knihovních tezaurů), v případě DCV vícerozměrně (např. pro potřebu statistických úřadů nebo podnikových datových skladů).

SKOS obsahuje mj. třídy `skos:Concept` (pro jednotlivý koncept) a `skos:ConceptScheme` (pro konceptové schéma jako množinu souvisejících konceptů). Přiřazení konceptu ke schématu zajišťuje vlastnost `skos:inScheme`. V rámci jednoho schématu jsou koncepty spojeny vlastnostmi vyjadřujícími hierarchii, zejména `skos:broader` a `skos:narrower`. Příkladem dat reprezentovaných pomocí SKOS je

```
ex:Savci      rdf:type         skos:Concept
ex:Savci      skos:broader     ex:Obratlovci
ex:Savci      skos:inScheme    ex:BiologickaTaxonomie
```

Pro DCV[19] je zásadní mj. třída `qb:Observation`, reprezentující abstraktní "pozorování" jako střed "hvězdice" (ev. "sněhové vločky") zachycující agregovaná data v dimenzionálním pohledu. Instance `qb:Observation` jsou spojené s daty reprezentujícími *dimenze*, *míry*, ev. *atributy* vícerozměrné datové kostky pomocí vlastností, které jsou pomocí predikátu `rdfs:subPropertyOf` podřazeny abstraktním vlastnostem `qb:DimensionProperty`, `qb:MeasureProperty` a `qb:AttributeProperty`. Každá instance je dále přiřazena určitému datasetu (`qb:Dataset`), a ten se pomocí vlastnosti `qb:structure` odvolává na tzv. *definici datové struktury* (DSD), kde jsou používané dimenzionální vlastnosti podrobněji specifikovány. Příkladem (upraveným ze specifikace DCV) takto vyjádřených dat je demografické zjištění z datasetu o očekávané délce života:

```
ex:dataset1      rdf:type         qb:DataSet
ex:dataset1      qb:structure     ex:dsd1
```

[15] `http://lov.okfn.org/dataset/lov/`
[16] `http://xmlns.com/foaf/0.1/`, entity dále s prefixem `foaf`.
[17] `https://www.w3.org/TR/skos-reference/`
[18] `https://www.w3.org/TR/vocab-data-cube/`
[19] Jako prefix se používá `qb`, odvozený ze slova "cube".

```
ex:observation1  rdf:type        qb:Observation
ex:observation1  qb:dataSet      ex:dataset1
ex:observation1  ex:refUzemi     ex:MoravskoslezskyKraj
ex:observation1  ex:refObdobi    ex:rok2016
ex:observation1  ex:pohlavi      ex-kod:pohlavi-M
ex:observation1  ex:jednotka     ex-kod:rok-R
ex:observation1  ex:delkaZivota  76.7
```

První dvě trojice definují celý dataset a jeho vztah k DSD. Zbylé se týkají jednoho pozorování popsaného pěti dimenzemi a jednou mírou (`ex:delkaZivota`), jejíž jednotka je upřesněna atributem (`ex:jednotka`). Všimněme si, že dimenze `ex:pohlavi` a atribut `ex:jednotka` mají hodnotu z jiného jmenného prostoru – jedná se o hodnotu z kódovníku, který typicky bývá reprezentován pomocí SKOS (jednotlivé kódy odpovídají konceptům podle SKOS a mohou být proto i hierarchicky uspořádány).

Na vývoji slovníků pro propojená data se v posledních letech podílel i autor a jeho tým. Jedná se zejména o slovník pro popis dat o veřejných zakázkách, *Public Contracts Ontology*,[20] [17], popřípadě o slovník o přístupnosti budov pro tělesně postižené, *Ontology of Building Accessibility*,[21] vyvinutý v rámci diplomové práce pod jeho vedením. Vede také tým VŠE účastnící se projektu EU OpenBudgets.eu, v rámci kterého vznikl model pro reprezentaci fiskálních dat[22] založený na slovnících DCV a SKOS (k modelu vytvořenému zejména doktorandem J. Mynarzem a několika dalšími kolegy autor sám přispěl jen formou občasné zpětné vazby).

Charakteristické strukturní odlišnosti výše uvedených typů ontologií jsou jedním ze vstupních zjištění motivujících originální výzkum popsaný v sekcích 4.1–4.4 tohoto textu.

3 Vybrané pojmy ontologického inženýrství

Ontologické inženýrství je celosvětově etablovanou disciplínou, zahrnující stovky přístupů a rozsáhlou terminologii. V tomto textu uvedeme jen několik klíčových pojmů potřebných pro další výklad. Zájemce o hlubší seznámení s problematikou odkazujeme na stěžejní monografii oboru *Ontological Engineering* [9], případně na editovanou knihu *Handbook on Ontologies* [24]. V českém jazyce je dostupná např. takto zaměřená kapitola 4 v knize *Umělá inteligence (6)* [31].

3.1 Přepoužití ontologií

Úlohu *přepoužití ontologií* můžeme rozdělit na dvě podúlohy:

1. přepoužívání na úrovni *ontologie – ontologie*: pojmy nově navržené ontologie jsou navázány na pojmy z existujících ontologií; zde může jít buď o *začlenění* částí existujících ontologií do nové nebo o vytvoření samostatně uchovávaného *mapování* mezi pojmy z existujících a z nové ontologie;

2. přepoužívání na úrovni *datová sada – ontologie*: nově publikovaná datová sada je popsána pomocí pojmů z existujících ontologií.

Do oblasti ontologického inženýrství striktně vzato spadá jen první podúloha. Druhá je však v praxi ještě významnější (počet nově vznikajících datových sad je větší než počet nově

[20] `http://purl.org/procurement/public-contracts`
[21] `http://w3id.org/charta77/jup`
[22] Popsaný v oficiálním výstupu projektu, `http://openbudgets.eu/assets/deliverables/D1.5.pdf`.

vznikajících ontologií), navíc rozdíl mezi nimi není ostrý, protože v případě přepoužití entit z existujících ontologií z nich vzniká implicitní *datové schéma* datové sady, které má ze strukturně-logického hlediska podobný charakter jako samostatná ontologie.

Konkrétní datová sada také často není popsána pojmy z jediné ontologie (datového slovníku), ale kombinuje odkazy na větší počet ontologií, často vzniklých nezávisle a potenciálně obsahujících vzájemné překryvy nebo nekonzistence. Užitečným prostředkem pro analýzu způsobu, jak jsou ontologie přepoužity v existujících datových sadách (de facto odhalení implicitního datového schématu) a následné navržení jeho přepoužití v nově vytvářených datech) je automatická *sumarizace* datových sad s následnou *vizualizací* souhrnů. Metoda sumarizace a následné vizualizace schématu datové sady s důrazem na odlišení různých zúčastněných ontologií byla vyvinuta ve spolupráci autora a jeho dvou doktorandů, M. Dudáše a J. Mynarze, a prezentována na konferenci ESWC 2015 [5].

3.2 Mapování ontologií

Mapování ontologií [7] je úloha spočívající v nalezení tzv. korespondencí, tj. vztahů ekvivalence, subsumpce (využití predikátu `rdfs:subClassOf`, ev. `rdfs:subPropertyOf`) nebo i jiných vztahů, mezi entitami (třídami nebo vlastnostmi) ze dvou, případně více ontologií. Může být prováděno ručně nebo (zejména v případě rozsáhlejších modelů) automaticky. Automatické metody se typicky opírají o míru lexikální shody názvů entit, a často i o logickou a grafově-teoretickou strukturu mapovaných ontologií. Rozvoj metod automatického mapování je již více než 10 let stimulován pravidelným konáním *Ontology Alignment Evaluation Initiative*, mezinárodní soutěže mapovacích systémů[23] ve vazbě na workshop *Ontology Matching* kolokovaný se světovou konferencí ISWC.

3.3 Ontologické vzory

Pojem "vzor" se v souvislosti s ontologiemi vyskytuje ve dvou významových odstínech. V ontologiích jako inženýrských artefaktech se opakovaně objevují (jak na úrovni jedné ontologie, tak i napříč různými) určité pravidelnosti, které můžeme obecně označit jako *ontologické vzory*, případně jako (ontologické) *empirické* vzory. Na druhou stranu v ontologickém inženýrství existuje snaha explicitně formulovat a opakovaně využívat, podobně jako v případě softwarového inženýrství, "obecná řešení často se vyskytujících problémů", tedy (ontologické) *návrhové* vzory.[24] Oba typy vzorů se do jisté míry prolínají, protože přítomnost empirických vzorů může být důsledkem využívání návrhových vzorů se strany tvůrců ontologie.

Ontologické návrhové vzory Návrhové vzory jsou již řadu let shromažďovány v rámci webových kolekcí, z nichž nejznámější je rozsáhlý portál *OntologyDesignPatterns.org* vzniklý v rámci projektu EU NeOn a katalog udržovaný Univerzitou v Manchesteru.[25] Nejnověji byly různé varianty návrhových vzorů a případů jejich užití popsány v editované knize *Ontology Engineering with Ontology Design Patterns*[26] [13]. *OntologyDesignPatterns.org* (i velká část odborné literatury) rozlišuje několik hlavních typů návrhových vzorů, z nichž vybíráme ty, které jsou relevantní v kontextu tohoto textu (zejména se omezíme na vzory týkající se ontologií v jazyce OWL):

- *Strukturní* (někdy též logicko-strukturní) vzory. Jejich základem je určitá kombinace logických konstruktů jazyka OWL; naopak neobsahují konkrétní ontologické entity

[23] Zatím poslední ročník viz `http://oaei.ontologymatching.org/2016/conference/`.

[24] Zřejmě nejcitovanější počáteční prací z této oblasti je článek A. Gangemiho z konference ISWC 2005 [8]. Stojí asi za zmínku, že článek cituje o rok starší, rozsahem skromnou práci autora tohoto textu [25], která budoucí rozmach výzkumu ontologických vzorů v mírném předstihu předjímala.

[25] `http://www.gong.manchester.ac.uk/odp/html/`

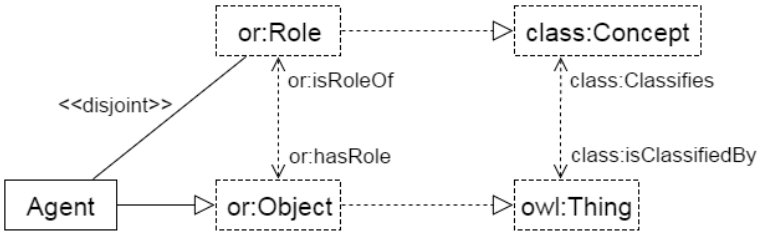
[26] Tým vedený autorem tohoto textu se na ní podílel kapitolou věnovanou využití vzorů pro transformace ontologií [32] (Kap. 12), a příspěvkem ke kolektivně formulovaným výzkumným otázkám oblasti (Kap. 9).

(namísto nich jsou uvažovány jen proměnné, za které lze entity dosazovat, vzor je tedy vlastně "šablonou" pro fragmenty konkrétních ontologií). Smyslem strukturního vzoru je popsat způsob, jak vyřešit určitý obecný modelovací problém, plynoucí z omezených vyjadřovacích možností jazyka. Příkladem je reprezentace vztahu s více než dvěma účastníky (tzv. "n-ární relace"), jakou je třeba vztah prodeje mezi prodávajícím, kupujícím a předmětem koupě.[27] Vzor v takovém případě (v jedné své variantě) doporučuje zavést novou třídu reprezentující pojem takového vztahu, a vlastnost/i spojující instance této "umělé" třídy s účastníky vztahu.

Množina tvrzení (šablona) ovšem nedává smysl sama o sobě, ale jen ve spojení s typovou modelovací situací, kterou má řešit. Tu bývá nutno popsat v textové podobě, a pokud možno doprovodit příklady. V tomto ohledu jsou velmi dobře rozebrané tři "rodiny" vzorů (pro použití třídy jako hodnoty vlastnosti, pro n-ární relace, a pro rozklady/množiny hodnot) zpracované skupinou pro ontologické inženýrství (OEP) při konsorciu W3C.[28]

- *Obsahové* vzory. Na rozdíl od strukturních vzorů se jedná o množinu tvrzení využívající konkrétní (byť zpravidla velmi obecné) ontologické entity. Jedná se vlastně o miniaturní ontologie popisující často se vyskytující situace, které lze jako celek přepoužít v nově vytvářených ontologiích, zpravidla tak, že se jim entity nové ontologie podřadí pomocí `rdfs:subClassOf` nebo `rdfs:subPropertyOf`. Pro podporu návrhu ontologií založeného na obsahových vzorech byla v institutu ISTC-CNR (Itálie) vyvinute agilní metodika XD [4].

  Jako příklad můžeme uvést vzor *AgentRole*, vyjadřující skutečnost, že určitý vědomý "agent" hraje určitou "roli". Vzor je publikován v rámci portálu *OntologyDesignPatterns.org*[29] a jeho struktura je zobrazena v notaci obdobné UML na Obr. 1. Vzor *AgentRole* (s prvky ohraničenými plnou čarou) využívá importovaný speciálnější vzor *ObjectRole*[30] (jeho entity jsou uvedeny prefixem `or`), a ten ještě dále importuje vzor *Classification*[31] (entity s prefixem `class`); entity obou importovaných vzorů jsou ohraničeny čarou přerušovanou. Pokud budeme postupovat od nejobecnějšího vzoru *Classification*, ten umožňuje pouze vyjádřit vztah mezi určitou entitou a konceptem, do kterého je tato entita "klasifikována"; jedná se o jistý druh opisu nám již známého instanciačného predikátu `rdf:type` (s tím, že cílová třída je zde metamodelována individuálem – instancí třídy `class:Concept`). Vzor *ObjectRole* pak již zavádí pojem "sehrávání role" jako speciálního druhu "klasifikace"; entitou, která sehrává roli, mohou být libovolné objekty (ovšem nikoliv např. vlastnosti). Konečně, vzor *AgentRole* nově zavádí specifičtější třídu objektů sehrávajících role, tedy (aktivních) agentů. Třída `Agent` je navíc explicitně deklarována jako disjunktní se třídou `or:Role`.



Obrázek 1: Obsahový vzor *AgentRole* a jeho importované vzory

[27] Na tento příklad se znovu odkazujeme v sekcích 3.4 a 4.3.
[28] `https://www.w3.org/2001/sw/BestPractices/OEP/`
[29] `http://ontologydesignpatterns.org/wiki/Submissions:AgentRole`
[30] `http://ontologydesignpatterns.org/wiki/Submissions:Objectrole`
[31] `http://ontologydesignpatterns.org/wiki/Submissions:Classification`

Ukázku využití vzoru *AgentRole* (ovšem ne při tvorbě nové ontologie, ale při transformaci ontologie existující), si předvedeme v sekci 4.2.

- *Mapovací* vzory. Podobají se strukturním vzorům (rovněž používají jen logické konstrukty OWL, s proměnnými místo konkrétních entit), ale jsou fakticky jejich nadstavbou. Jejich vnitřní struktura je členěna na dva strukturní vzory (každý identifikovaný v jedné z mapovaných ontologií) a na šablonu mapovacích tvrzení. Na rozdíl od strukturních a obsahových vzorů nejsou určeny pro návrh ontologií, ale pro návrh mapování mezi nimi. Mapovací vzory mají velmi blízko k *transformačním* vzorům (ty v tomto chápání, pro transformaci z OWL do OWL, na *OntologyDesignPatterns.org* uvažovány nejsou), kterým se podrobněji věnuje sekce 4.2. Přímo mapovacími vzory se zabýval O. Zamazal v rámci své disertační práce (pod vedením autora), viz sekce 4.1.

Ontologické empirické vzory Analýza *empirických vzorů* má zpravidla charakter automatického dolování z dat, i když nelze zanedbat i podíl "ruční" analýzy (nad menšími vzorky ontologií), která je pro některé komplexnější vzory jedinou možností. Do úvahy se zpravidla berou kombinace strukturních i lexikálních charakteristik. Mezi nejvýznamnější práce z této oblasti patří analýza pravidelností v biomedicínských ontologiích autorů z Univerzity v Manchesteru [16], využívající shlukování, a z Technické univerzity v Poznani, založená na asociačních pravidlech [14].

Do kategorie analýzy empirických vzorů spadají i některé práce autora tohoto textu; ten se spolu s kolegy zaměřil na sledování souvislostí mezi názvy tříd a vlastností spojených hierarchickými [38] i nehierarchickými (zvl. 'domain' a 'range') [30] vztahy. Některé z výstupů této analýzy (strukturně-lexikální vzory) byly následně využity pro úlohu opravování "názvových" chyb v ontologiích, řešenou ve spolupráci s kolegy z Univerzity v Lipsku [40], a pro úlohu začlenění obsahového vzoru do existující ontologie, podrobněji popsanou v části 4.2. Specifická kategorie empirických vzorů vznikajících nad mapováním ontologií, rovněž studovaná autorovým týmem, je dále zmíněna v sekci 4.1.

3.4 Strukturní heterogenita ontologií

Na problém řešený pomocí strukturních návrhových vzorů se můžeme podívat i z odlišného hlediska. Vezměme si za příklad zmíněný vztah "prodeje". Ten totiž můžeme chápat i jako binární, např. v případě, že kupující ještě není znám; v takovém případě ho lze vyjádřit pomocí objektové vlastnosti, jejíž definiční obor bude, dejme tomu, třída všech osob, a oborem hodnot třída všech produktů. Jedna ontologie (odlišíme ji prefixem `o1`) tedy může situaci zachycovat takto

```
o1:prodava rdfs:domain o1:Osoba
o1:prodava rdfs:range o1:Produkt
```

a druhá (s prefixem `o2`), využívající reprezentaci vztahu pomocí třídy ve stylu "n-ární relace", takto

```
o2:prodejce rdfs:domain o2:Prodej
o2:prodejce rdfs:range o2:Osoba
o2:predmetProdeje rdfs:domain o2:Prodej
o2:predmetProdeje rdfs:range o2:Produkt
```

Využití každé z ontologií tedy vede k odlišné struktuře dat pro realitu, která je v obou případech shodná. Povšimněme si přitom, že obě varianty mají své výhody i nevýhody. První varianta je úspornější a přehlednější: konkrétní tvrzení je tvořeno jedinou trojicí, kterou lze za jistých okolností přečíst téměř jako větu v přirozeném jazyce, např.

```
ex:JanNovak o1:prodava ex:OsobniVuz_RZ_1AC23456
```

Neumožňuje však přidat do vztahu dalšího aktéra (kupujícího), ani cenu či jiné (časové, místní, apod.) okolnosti prodeje.

Jiným příkladem může být modelování hierarchie. V sekci 2.3 jsme se setkali se způsobem modelování hierarchie pomocí predikátů slovníku SKOS:

```
ex:Savci      skos:broader     ex:Obratlovci
```

Stejný hierarchický vztah můžeme ovšem vyjádřit i pomocí `rdfs:subclassOf`:[32]

```
ex:Savec      rdfs:subClassOf  ex:Obratlovec
```

Opět lze identifikovat výhody i nevýhody obou řešení. V případě použití struktury SKOS máme např. možnost (pomocí `skos:inScheme`) explicitně vyjádřit, že pojem `ex:Savci` je součástí schématu `ex:BiologickaTaxonomie`, zatímco v případě použití `rdfs:subClassOf` bychom podobné přiřazení pro *třídu* `ex:Savci` realizovali poměrně komplikovaně. Naopak bychom však získali možnost reprezentovat jednotlivá fyzická zvířata jako instance této třídy a pomocí logického odvozování je identifikovat jako obratlovce. Toto by zase bylo obtížnější v reprezentaci SKOSovské, přičemž automatická konstrukce taxonomie z modulárních definic (jakou jsme si ukázali na příkladu z biomedicínské ontologie, rovněž v sekci 2.3) by pro SKOSovské koncepty nebyla možná vůbec.

Demonstrované příklady variantní reprezentace nejsou umělé, ale v praxi se velmi často vyskytují. V různých oblastech mohou být ontologie pojednávající o stejných "jsoucnech" různě strukturované, což je v některých případech dáno jen náhodným rozhodnutím tvůrce ontologie resp. modelovaných dat, častěji je však oprávněným důvodem to, že výhody určité varianty pro daný případ užití převažují nad nevýhodami. Můžeme proto hovořit o *strukturní heterogenitě* ontologické vrstvy sémantického webu, s tím, že

- různé způsoby vyjádření téže reality odpovídají různým *strukturním vzorům*
- a souvislost mezi takovými alternativními strukturními vzory můžeme postihnout pomocí *mapovacích vzorů*.

V sekcích 4.1 a 4.2 se proto budeme věnovat mapovacím vzorům, a dále *transformačním vzorům*, které s nimi mají úzkou souvislost. Předtím však ještě stručně představíme konkrétní aktivitu iniciovanou autorem tohoto textu, která měla na cíl poskytnout materiál pro studium strukturní hetergenity ontologií jako takové.

Kolekce OntoFarm Autor vyšel z předpokladu, že pro detailní studium strukturní heterogenity, vedoucí k odhalení reprezentativní množiny vzorů, je potřeba získat rozsáhlejší kolekci ontologií, které budou modelovat stejnou věcnou doménu, avšak vzniknou nezávisle na sobě. Proto v r. 2005 vznik takové kolekce inicioval, s tím, že jako pro akademickou obec dobře srozumitelný předmět modelování zvolil "organizaci konferencí". Zárodek kolekce nazvané *OntoFarm* (4 ontologie) byl prezentován na konferenci ISWC 2005 [34] a v dalších letech postupně rozšiřován (v r. 2006 již zahrnoval 10 ontologií, v r. 2007 14 ontologií, a aktuálně jich obsahuje 16). Již od počátku se na správě a publikaci kolekce podílel student O. Šváb, který postupně péči o ni zcela převzal. Tvůrci ontologií byli převážně studenti magisterského studia (v rámci předmětu zaměřeného na technologie sémantického webu), v menší míře akademičtí pracovníci. Vstupním materiálem pro modelováním jednotlivých ontologií byly nejčastěji *softwarové nástroje* pro organizaci konferencí, méně často *osobní zkušenost* jejího tvůrce s organizováním konference, případně (jen) *webové stránky* určité konference. Kolekce vzbudila ohlas zejména v komunitě mapování ontologií, a již od r. 2006

[32] Všimněme si, že jednou z lexikálních konvencí OWL je uvádění názvů tříd v jednotném třídě, tj. jako označení jednotlivé instance dané třídy. Hierarchie SKOS se takovou konvencí neřídí, i proto, že zachycené pojmy nemusí vždy odpovídat třídám majícím instance.

byla používána pro jednu ze sekcí soutěže OAEI zmíněné v sekci 3.2. Překladem lexikální složky 7 konferencí OntoFarm z angličtiny do 8 dalších jazyků (včetně češtiny – překlad O. Šváby-Zamazala a V. Svátka) pak vznikla nová, vícejazyčná kolekce *MultiFarm* [15], určená pro testování mapovacích systémů napříč jazyky. Pro ni byla založena samostatná sekce OAEI.[33]

Souhrnná studie popisující OntoFarm a její využívání, vyšlá (preprint, leden 2017) v *J. Web Semantics* [42], mj. shrnuje, že kolekce byla využita v rámci osmi mezinárodních a řady národních projektů a odkazuje se na ni přibližně 80 vědeckých publikací, převážně zaměřených na mapování ontologií, ale i na další subdisciplíny ontologického inženýrství.

4 Vybrané scénáře a metody přepoužití ontologií

Přestože již předchozí sekce se na řadě míst odvolávala na výzkumný přínos autora k problematice, těžiště originálních výsledků sumarizovaných v tomto textu je až v sekci současné. Jednotícím prvkem čtyř zde uvedených scénářů je úsilí o usnadnění nebo zkvalitnění přepoužívání ontologií na sémantickém webu. Problémem, který je přitom nutno překonávat, je vesměs strukturní heterogenita ontologií rozebraná v sekci 3.4.

4.1 Přepoužití ontologií pomocí komplexních mapovacích vzorů

Tradiční přístupy k mapování ontologií jsou založeny na rozpoznávání vztahu ekvivalence (`owl:equivalentClass`), případně subsumpce (`rdfs:subClassOf`) mezi pojmenovanými třídami ze dvou ontologií. Tvrzení typu ekvivalence a subsumpce, vždy s proměnnou zastupující pojmenovanou třídu na každé straně, jsou tedy nejjednoduššími mapovacími vzory. Mapování, kde (přinejmenším) jeden z mapovaných konceptů může být složeným výrazem, ovšem vyžaduje zachycení *komplexními* mapovacími vzory. Ty jsou "přikládány" každou svou stranou k jedné z porovnávaných ontologií (entity z ontologií jsou přitom dosazovány za proměnné ze vzorů), a pokud je na obou stranách dosazení dostatečně přesné, vygeneruje se konkrétní případ korespondence. Vzhledem ke komplexitě úlohy (možných "přiložení" každého vzoru je velké množství) se předpokládá poloautomatická aplikace řízená uživatelem v rámci grafického mapovacího rozhraní.

První ucelený katalog takových vzorů poprvé představil v r. 2007 na workshopu *Ontology Matching* F. Scharffe (Univ. Innsbruck, Rakousko, později INRIA, Francie) s kolegy [21]. Příkladem komplexního mapovacího vzoru je tzv. "class by attribute correspondence", kdy pojmenovaná třída A z jedné ontologie odpovídá ve druhé ontologii konjunkci pojmenované třídy B a existenční restrikce nad vlastností R s hodnotou specifikovanou jako jednoprvková třída s prvkem i. Konkrétní výskyty tohoto vzoru odpovídají logickým tvrzením OWL ve struktuře

$$A \equiv B \sqcap \exists R.\{i\}$$

například, pokud máme jednu ontologii obsahující taxonomií vín a jinou zachycující vinařské oblasti, konkrétní korespondencí může být:

$$BordeauxkeVino \equiv Vino \sqcap \exists\ vinarskaOblast\ .\ \{Bordeaux\}$$

Některé jiné vzory ze Scharffova katalogu ovšem zahrnovaly i konstrukce nepodporované v rámci tvrzení OWL, např. *konkatenaci* řetězcových hodnot. Příkladem výskytu takového vzoru je mapování nám již známé ontologie FOAF na rovněž populární ontologii *vCard*:[34] první z nich modeluje jméno člověka jako celek (řetězcovou datovou vlastností `foaf:name`), zatímco druhá používá vlastnost objektovou (`vc:name`), jejíž hodnotou je abstraktní entity odkazující na tři složky jména pomocí tří samostatných vlastností (`vc:family-name`,

[33] `http://oaei.ontologymatching.org/2016/multifarm/`
[34] `https://www.w3.org/TR/vcard-rdf/`

`vc:given-name` a `vc:additional-name`). Hodnota `foaf:name` je tedy mapována na zřetězení hodnot tří zmíněných vlastností.

K poněkud odlišnému pojetí mapovacích vzorů dospěli O. Šváb a V. Svátek v rámci vyhodnocování výsledků mapování ontologií z kolekce OntoFarm v rámci OAEI, již počínaje rokem 2006. Zde se jednalo o mapovací vzory spíše na *makro-úrovni*, tj. agregující více elementárních mapování. Jednoduchým příkladem je situace, kdy je třída A z ontologie O_1 některými mapovacími nástroji určena jako ekvivalentní třídě B z ontologie O_2 a některými jako ekvivalentní třídě C, rovněž z O_2, přičemž v rámci O_2 platí $C \sqsubseteq B$. Vidíme, že z pohledu typologie ontologických vzorů jde o vzory spíše *empirické* než návrhové. Výsledky agregované analýzy (dolování asociačních pravidel z dat) nad takovými vzory spolu s dalšími rysy získanými z dat o výsledcích OAEI byly představeny na evropské konferenci ESWC 2007 [36].

Na okraj poznamenejme, že O. Šváb později v rámci své disertační práce (zpracovávané pod vedením autora) navázal dlouhodobou spolupráci s F. Scharffem v oblasti mapovacích vzorů charakteru logických tvrzení. Ta vyvrcholila jejich společnou publikací v časopise *Knowledge and Information Systems* [22] věnované rozšířenému katalogu mapovacích vzorů a jejich evaluaci; na této fázi výzkumu se ovšem autor již nepodílel.

Pokud jde o podporu přepoužívání ontologií, vzory ze Scharffova katalogu jsou přímým nástrojem pro tvorbu mapování, které přepoužívání ontologií umožňuje. Empirické vzory analyzované Švábem a Svátkem oproti tomu poskytovaly materiál pro zdokonalování mapovacích nástrojů, které se soutěže OAEI účastnily. Jejich přínos pro přepoužitelnost ontologií byl proto pouze nepřímý.

4.2 Začlenění ontologie/vzoru do ontologie podpořené transformací

Ještě než představíme případ přepoužití samotný, uvedeme si obecnou technologii transformace ontologií s využitím transformačních vzorů, která se v jeho rámci používá.

Transformace ontologií s využitím transformačních vzorů je originálním postupem navrženým a prakticky uplatňovaným autorem a jeho týmem od r. 2009.[35] Jedná se o obdobu mapování ontologií pomocí komplexních mapovacích vzorů popsaného v části 4.1; rozdíl spočívá zejména v tom, že na vstupu je jen jediná ontologie, zatímco druhá ontologie (nebo jen dílčí varianta vstupní ontologie) vzniká teprve její transformací.

Transformační vzor se skládá ze *zdrojového ontologického vzoru*, *cílového ontologického vzoru*, a *popisu transformace* převádějící výskyt zdrojového vzoru na strukturu vzoru cílového. Popis transformace má komponentu *logicko-strukturní*, vyjadřující převod struktury entit, a *lexikální*, vyjadřující transformaci pojmenování entit (lexikální vzory se používají i v rámci zdrojového vzoru, kde spolurozhodují o tom, zda lze transformaci vůbec aplikovat). Popisný formalismus problematiky byl poprvé publikován ve sborníku evropské konference EKAW 2010 [39].

Podobně jako v případě komplexního mapování se u transformace ontologií založené na vzorech předpokládá poloautomatický, interaktivní scénář využití:

1. uživatel (ontologický inženýr) zvolí potenciálně relevantní transformační vzor
2. transformační systém najde korektní "přiložení" jeho levé strany ke vstupní ontologii, v podobě seznamu n-tic dosazení za proměnné vzoru
3. uživatel v případě potřeby odstraní nežádoucí n-tice ze seznamu

[35] V letech 2010-2012 s podporou takto zaměřeného projektu GAČR č.P202/10/1825, "Automation of Ontology Pattern Detection and Exploitation" (PatOMat); souhrnná webová stránka projektu je `http://patomat.vse.cz`.

4. systém provede transformaci a zobrazí rozdíl mezi původní a transformovanou ontologií (vstupní entity mohou být v nové verzi ontologie ponechány, zpravidla i propojeny s novými entitami, nebo mohou být naopak odstraněny)

5. transformovaný výsledek je poté možno uložit pod novým názvem.

Vytvořená softwarová podpora[36] zahrnuje jádrový *soubor transformačních služeb* v podobě RESTovských služeb i Javovské knihovny (vytvořil O. Zamazal), interaktivního grafického uživatelského rozhraní GUIPOT pro *realizaci transformací*, a rovněž grafického *editoru transformačních vzorů* (oboje vytvořil doktorand M. Dudáš). Software je vesměs přístupný na webu `http://owl.vse.cz:8080/patomat/tools.html`; tamtéž jsou uvedeny i příklady vytvořených transformačních vzorů pro různé úlohy.

Vyvinutý postup a nástroje byly postupně použity na široké spektrum úloh ontologického inženýrství. Jejich přehled byl publikován v časopise *Computing and Informatics* [41], zde si je zmíníme jen stručně (s výjimkou úlohy odpovídající vybranému scénáři profesorské přednášky); většina použitých vzorů je dostupná v rámci katalogu[37] dostupného z projektového webu.

- Transformace stylu *OWL vs. SKOS*. Jedná se o situaci řešící druhý z případů strukturní heterogenity uvedené v sekci 3.4. Určitá část hierarchie tříd OWL, ze které chceme udělat kódovník, je s pomocí jednoduchého vzoru automaticky převedena do podoby taxonomie SKOS. Transformace se provádí rekurzivně, tj. postačí označit třídu ontologie, jejíž všechny, i nepřímé, podtřídy se stanou koncepty SKOS. Obdobně (s pomocí "opačně orientovaného" vzoru) lze naopak z taxonomie SKOS vytvořit ontologii v OWL; sémantickým předpokladem v tom případě ovšem je, aby vstupní taxonomie respektovala množinovou sémantiku (predikáty typu `skos:broader` totiž mohou odpovídat nejen vztahu obecnějšího a speciálnějšího typu objektů, ale také vztahu celku a části, případně může podřazený koncept již odpovídat jednotlivému objektu a nikoliv typu objektů).

- Transformace do *jednoduššího dialektu jazyka*. Některé konstrukce jazyka OWL mohou způsobovat problémy při odvozování nástroji, které je nepodporují. Východiskem je jejich transformace na konstrukce jiné, povolené, které ty původní aproximují. Ve spolupráci s Univerzitou v Mannheimu byly vytvořeny a aplikovány vzory transformující např. třídy tvořené výčtem svých instancí [37].

- Refaktoring *lexikálního aspektu* ontologie. Častou překážkou srozumitelnosti ontologií je neúplnost pojmenování entit. Příkladem je situace, kdy tvůrce ontologie nazve podtřídu třídy *Auto* pouze *Osobni*. Zatímco při prohlížení ontologie v hierarchickém pohledu je význam z kontextu jasný, v případě zobrazení třídy v jiné souvislosti se může význam ztrácet – např. pokud by pro určitou pracovní pozici bylo požadováno, aby osoba byla ve vztahu *jeVlastnikem* k instanci zmíněné třídy *Osobni*. Aplikace jednoduchého transformačního vzoru může zajistit automatické doplnění tříd, jejichž název je přídavným jménem, o podstatné jméno, které je z lingvistického hlediska "hlavním" prvkem názvu jejich nadtříd. Tato funkce byla v rámci spolupráce v projektu EU LOD2 kolegy z Univerzity v Lipsku implementována do jejich nástroje ORE a společně provedeny experimenty [29].

- *Mapování ontologií*. Úlohu transformace ontologií můžeme aplikovat jako alternativu v situaci, kdy bychom pro mapování ontologií museli použít komplexní mapovací vzory. Jednu z ontologií (tu, na jejíž straně se vyskytuje složený výraz) můžeme totiž transformovat s pomocí transformačního vzoru (strukturně víceméně odpovídající vzoru

[36] Na evropské konferenci EKAW 2012 získala tato sada nástrojů ocenění jako "nejlepší demo" [35].
[37] `http://nb.vse.cz/~svabo/patomat/tp/new/index.html`

mapovacímu), a výslednou ontologii již následně automaticky mapovat na druhou ontologii s pomocí jednoduchých, převážně lexikálních metod. Na výzkumu se vedle O. Šváb-Zamazala a autora opět podílel i F. Scharffe, a další odborník z INRIA J. David [43].

- Transformace ontologie jako *adaptace* na ontologii "nejlepší praxe" nebo návrhový vzor do ní začleněný.

Tento poslední scénář rozebereme samostatně, jednak proto, že více než předchozí klade důraz na *přepoužívání ontologií*, jednak s ohledem na výraznější přímý podíl autora na jeho realizaci, včetně experimentální části. Ukážeme si na něm některé obecné aspekty celého přístupu: (variantní) konstrukci levé a pravé strany transformačního vzoru, i lexikální prvek transformace.

Podstatou řešeného problému je skutečnost, že tvůrce ontologie nemusí vždy ve fázi její tvorby mít povědomí o předchozího existenci modelu (tzv. "artefaktu nejlepší praxe", zkratka BPA), který by bývalo vhodné přepoužít jako jednu z jejích obecných částí. Tento model je často následně identifikován až ve fázi, kdy byla ontologie vytvořena. Zpětná integrace modelu do již vytvořené ontologie pak může vyžadovat určité její strukturní přizpůsobení. Takové přizpůsobení je možné běžnými technologiemi realizovat dvěma způsoby: buď *ručním editováním*, nebo napsáním jednoúčelového skriptu v *programovacím jazyce*, který úpravy provede hromadně. Technologie transformací založených na vzorech zde poskytuje "střední cestu": přizpůsobení může realizovat i ontologický inženýr, který není programátorem, v grafickém prostředí, na které je zvyklý (grafické rozhraní GUIPOT je realizováno jako zásuvný modul do populárního editoru Protégé[38]), a s výrazně menším objemem ruční práce, než v případě editování "tvrzení po tvrzení".

"Artefaktem nejlepší praxe" může být základní ontologie, jádrové ontologie určitého oboru, nebo ontologický obsahový vzor. Zde si stručně rozebereme adaptaci ontologií na využití *obsahového vzoru* AgentRole představeného v sekci 3.3.

Ontologie vytvořená bez explicitního modelování *rolí* je může nepřímo zachycovat dvěma způsoby:

- Jako *třídy*, které jsou podtřídami určité třídy "agentů", např. třídy[39] *Teacher* nebo *Author*, fakticky vymezující časově omezenou roli (a nikoliv pevně danou množinu osob), budou podtřídami třídy *Person*.

- Pomocí objektové (případně datové) *vlastnosti*, která má třídu "agentů" nebo její některou podtřídu vymezenou jako svůj definiční obor (`rdfs:domain`), např. *teaches* nebo *wrote*.

Pokud chceme takovou ontologii adaptovat pro začlenění vzoru AgentRole, potřebujeme dvě varianty *levé strany* transformačního vzoru: pro role vyjádřené pomocí tříd a pomocí vlastností.

Pokud jde o *pravou stranu* transformačního vzoru, pro vstupní role vyjádřené pomocí *tříd* zde máme k dispozici dvě základní varianty:

- První varianta, kterou označíme jako "styl OWL", předpokládá, že pro každou vstupní třídu na výstupu vznikne opět *třída* rolí; např. pro třídu *Author* můžeme vytvořit třídu *AuthorRole*. Za předpokladu, že původní třídu ponecháme, v rámci transformace ji také spojíme s novou třídou pomocí existenční restrikce: $Author \equiv \exists hasRole.AuthorRole$

- Druhou variantu označíme jako "styl LOD" (protože odpovídá praxi používané v rámci otevřených propojených dat – "linked open data"); každé vstupní třídě bude odpovídat

[38] `http://protege.stanford.edu`

[39] V této části textu používáme anglické názvy tříd, protože na českých nelze dobře demonstrovat transformace názvů entit.

individuál vyjadřující příslušnou roli. Opět můžeme závislost původní třídy na vytvořené roli vymezit pomocí existenční restrikce, v tomto případě však její hodnotou bude výčtová třída s jedním prvkem (jde o tzv. restrikci na hodnotu, "value restriction"): $Author \equiv \exists hasRole.\{AuthorRole\}$

V obou případech se hierarchická struktura tříd (implicitních rolí) transformuje na explicitní hierarchii rolí. Ta bude v první variantě provázána pomocí *rdfs:subClassOf*, zatímco ve druhé variantě pomocí vyčleněné objektové vlastnosti nazvané např. *subRoleOf* (jde o stejný případ jako alternativa mezi hierarchií *rdfs:subClassOf* a strukturou SKOS popsaná v sekci 3.4).

V případě vstupní role vyjádřené pomocí *vlastnosti* musíme ještě vyřešit problém *pojmenování* třídy rolí, která z vlastnosti vznikne. Pro angličtinu lze navrhnout jednoduché lexikální vzory založené na prosté konkatenaci řetězců, které ve většině případů postačují pro srozumitelnost výsledného názvu. Tím může v daném případě být konkatenace jména třídy agentů (`?A`), transformované vlastnosti (`?r`), třídy v jejím oboru hodnot (`?C`), a fixních řetězců:

```
Role_of_+?A+_that_+?r+?C
```

Vzniklá třída rolí se pak může jmenovat např. *Role_ of_ Person_ that_ authorOf_ Document* (pro vlastnost *authorOf*, jejíž jméno je založeno na podstatném jménu s předložkovou vazbou), nebo *Role_ of_ Human_ that_ Supervises_ Activity* (v tomto případě je jméno vlastnosti slovesem ve 3. osobě j.č., což je pro lexikální transformaci ještě příznivější). Ještě lepšího výsledku by bylo možné dosáhnout s pomocí tzv. derivačně souvisejících slovních tvarů dostupných např. v tezauru WordNet[40]: takto bychom mohli pro vlastnost *supervises* automaticky vygenerovat přesně nazvanou roli: *SupervisorRole*.

Přístup byl empiricky otestován na 16 ontologiích z kolekce OntoFarm popsané v sekci 3.4, s využitím souboru nástrojů projektu PatOMat a výše popsané trojice transformačních vzorů (dvou pro transformaci tříd a jednoho pro transformaci vlastností). Ručním rozborem bylo ověřeno, že ze 154 rolí pro vstupní třídy bylo korektně vytvořeno 141 rolí (90%) odpovídajících AgentRole a ze 167 rolí pro vstupní vlastnosti bylo korektně vytvořeno 130 rolí (83%) odpovídajících AgentRole. Neúplnost transformace byla nejčastěji způsobena tím, že se ve vstupní ontologii vyskytlo složitější schématové tvrzení, které nebylo jednoduchými vzory podchyceno, např. obor vlastnosti byl vyjádřen jako disjunkce více tříd. Další příčinou byla skutečnost, že dvě z ontologií již v rámci svého vlastního jmenného prostoru používaly explicitní třídu "Role", nové třídy pro role pak sice byly vygenerovány korektně, ale byly následně podřazeny této třídě a ne třídě z AgentRole odpovídající nejlepší praxi a zajišťující interoperabilitu aplikací.

Studie zahrnující jak zmíněné experimenty s převodem na modelování pomocí rolí (realizované přímo autorem), tak i odlišný případ užití stejného postupu, adaptaci *produktových ontologií* na referenční ontologii elektronického obchodu GoodRelations (primárně realizovaný M. Dudášem), byla v r. 2016 publikována v časopise *J. Web Semantics* [26]. Již dříve byla jednoduchá varianta transformace pro AgentRole integrována do vývojové verze softwarové aplikace XDTools[41] vyvinuté partnery z ISTC-CNR.

4.3 Využití modelů ontologického pozadí

V sekci 3.4 věnované sémantické heterogenitě jsme si ukázali příklady alternativních vyjádření stejné skutečnosti v rámci jednoho ontologického jazyka, OWL. Tato vyjádření však zpravidla nejsou zcela rovnocenná z hlediska "věrnosti realitě", pokud jde o rozlišení jednotlivých *objektů* a jejich *typů*, nebo o rozlišení samostatně existujících *objektů* a jejich vzájemných *vztahů* (jejichž existence je podmíněna existencí příslušných objektů). Pokud

[40] `https://wordnet.princeton.edu/wordnet/`
[41] `http://neon-toolkit.org/wiki/XDTools`

např. uvažujeme vztah prodeje skutečně jen jako *vztah* prodávajícího, předmětu prodeje, ev. kupujícího a ceny (a ne např. jako *událost* prodeje konanou v určitém místě a čase), její vyjádření formou instance třídy *Prodej* a "připojovacích" vlastností *prodejce*, *predmetProdeje* apod. zavádí do ontologie úroveň komplexity, která v realitě fakticky neexistuje. Naopak, když ve druhém příkladě vyjadřujeme pojem *savce* jako ontologické *individuum* (začleněné do hierarachie SKOS), je to oproti realitě nepřesné, protože se jedná o pojem reprezentující rozsáhlou množinu fyzických živočichů.

Naskýtá se proto otázka, jestli by bylo možné jeden ze způsobů modelování zvolit jako "základní" a ostatní z něj odvozovat. Toto ale není schůdné, pokud se omezíme vyjadřovacími prostředky jazyka OWL:

- Jak už jsme naznačili v sekci 3.3, v pasáži věnované strukturním vzorům, OWL podporuje pouze binární relace; *n-ární relaci* v něm přímo vyjádřit nelze.
- Třída v OWL nemůže být zároveň instancí jiné třídy – deskripční logika, která mu dává formální základ, je totiž podmnožinou predikátové logiky prvního řádu, a proto nemůže obsahovat třídy (tj. unární predikáty ve smyslu prvořádové logiky) *vyšších řádů*.

Řešením tohoto problému je vytvoření nového ontologického reprezentačního jazyka, ve kterém budou zmíněná omezení uvolněna; současně by však měl být co nejpodobnější OWL, aby byl snadno pochopitelný vývojářům zvyklým na OWL a aby bylo možné jeho struktury do OWL co nejpřímočařeji transformovat. Autor ve spolupráci s M. Vacurou a kolegy z UK Bratislava (podepřené česko-slovenským mobilitním projektem LAAOS) takový jazyk, nazvaný *PURO*, v r. 2012 navrhl. Akronym PURO jednak evokuje snahu o "čistší" modelování než umožňuje OWL, jednak odkazuje na iniciály hlavních "ontologických distinkcí", se kterými operuje: jde o rozdíl mezi tzv. *jednotlivinami* a *obecninami* (P="particulars" vs. U="universals") a rozdíl mezi *objekty* a jejich *vztahy* (R="relationships" vs. O="objects"). Zvýšení vyjadřovací síly oproti OWL spočívá v tom, že obecniny (v PURO označované jako $\mathcal{B}$-typy) mohou být samy instancemi typů vyšších řádů, a dále v tom, že vztahy ($\mathcal{B}$-vztahy) v PURO mohou mít více účastníků než dva. Svou obdobu v PURO mají i datové vlastnosti OWL (s literálovými hodnotami), tzv. $\mathcal{B}$-valuace; ty se ovšem používají pouze pro vyjádření *kvantitativních* vlastností, a ne např. jako kódovníkové hodnoty. Je to proto, že pojmy reprezentované v OWL jako 'kódy', sémanticky téměř vždy odpovídají typům (popřípadě individuálním objektům) a jsou proto v PURO vyjádřeny jako $\mathcal{B}$-typy (případně jednotliviny – $\mathcal{B}$-objekty).

Modely PURO nejsou primárně určeny pro přímé využití v odvozovacích systémech nebo pro dodávání sémantiky rozsáhlým datovým zdrojům, ale mají sloužit jako podpůrný prostředek pro tvorbu nebo zlepšování ontologií vyjádřených v OWL (případně, jiném, strukturně obdobném jazyce). Proto je označujeme jako *modely ontologického pozadí* (zatímco na ně navázané ontologie OWL pak modelují ontologické popředí); symbol $\mathcal{B}$ v označení primitiv PURO odpovídá právě výrazu "background". Jedná se o pojem nově zavedený, který byl ovšem autorem zpětně aplikován i na etablovaný model *OntoClean* [11], jehož účel je obdobný jednomu ze scénářů využití PURO, ověřování sémantické koherence ontologie (ovšem s odlišnou sadou ontologických distinkcí).

Časová závislost vzniku modelu PURO a ontologií OWL může být dvojí:

- Model PURO může být abstrahován nad existující ontologií OWL, formou *anotací* přiřazených jednotlivým entitám této ontologie.
- Model PURO může být navržen primárně, a základ ontologie OWL z něj může být následně *vygenerován*.

V prvním případě se bude zpravidla jednat o ruční činnost ontologického inženýra, protože spolehlivé rozpoznání distinkcí PURO pravděpodobně přesahuje možnosti strojové ana-

lýzy. Poznamenejme, že i pro člověka se jedná o relativně intelektuálně náročnou činnost: pro množinu 20 tříd (reprezentativně vybraných ze tří populárních ontologií) dosáhla skupina 13 studentů znalostního inženýrství, na intuitivním základě, jen 57% shody s konsensuálním názorem dvou expertů (konkrétně, spolutvůrců PURO V. Svátka a M. Vacury). Na druhou stranu, vzájemného konsensu expertů bylo dosaženo pro 98% (92 ze všech 94) tříd z těchto tří ontologií.[42] V případě využití srozumitelného návodu s příklady lze předpokládat zlepšení kvality anotace i u laiků. Pro podporu úlohy anotování pomocí modelů pozadí byl dále v rámci diplomové práce S. Serry [23] vytvořen softwarový nástroj *B-Annot*, jako zásuvný modul do ontologického editoru Protégé. B-Annot umožňuje přiřadit distinkci PURO libovolné entitě v ontologii, navíc podporuje i anotování podle výše zmíněného alternativního modelu pozadí OntoClean.

Ve druhém případě bude model PURO opět zpravidla vytvářen ručně, ve vizuálním editoru. Vygenerování kódu OWL lze pak zajistit automatickými prostředky (transformačními vzory) obdobnými těm z projektu PatOMat.

Pro jazyk PURO bylo od té doby naformulováno několik obecných scénářů využití, z nichž většina má souvislost s *přepoužíváním ontologií*; na implementaci a testování většiny z nich se od r. 2014 pod vedením autora intenzivně podílí doktorand M. Dudáš.

- Prvním ze zkoumaných scénářů je *testování sémantické koherence* ontologie (na úrovní distinkcí PURO), popsaná v příspěvku na workshopu OWLED 2013 [27]. Jedná se o ověření, že 1) jeden konstrukt nespadá současně pod více neslučitelných typů, např. není současně $\mathcal{B}$-objektem i $\mathcal{B}$-typem, a že 2) $\mathcal{B}$-typy i $\mathcal{B}$-relace (což jsou obecné typy $\mathcal{B}$-vztahů) obsahují instance ($\mathcal{B}$-objekty resp. $\mathcal{B}$-vztahy) stejného řádu. Ověření se provádí pomocí metamodelování zkoumané ontologie v prostředí *meta-ontologie* PURO, která obsahuje integritní omezení potřebná pro testování koherence.[43] Triviálním příkladem, na kterém lze demonstrovat detekci nekoherence, je široce využívaná jádrová ontologie pro elektronický obchod, *GoodRelations*.[44] Ta mj. obsahuje obecnou třídu *ProductOrService* s podtřídami *Individual* (jejími instancemi jsou jednotlivé "kusy" produktu), *ProductOrServiceModel* (jejími instancemi jsou "katalogové" typy produktů) a *SomeItems* (každá její instance zastupuje libovolně velkou množinu "kusů"). Třídu *Individual* budeme jednoznačně modelovat jako třídu $\mathcal{B}$-objektů zatímco třídu *ProductOrServiceModel* jako třídu i $\mathcal{B}$-typů. S využitím deskripční logiky se pak odvodí, že třída *ProductOrService* je zároveň $\mathcal{B}$-typem 1. řádu (protože obsahuje $\mathcal{B}$-objekty, které mu dodá podtřída *Individual*) a zároveň $\mathcal{B}$-typem 2. řádu (protože obsahuje $\mathcal{B}$-typy 1. řádu, které mu dodá podtřída *ProductOrServiceModel*). S ohledem na "typově smíšenou" třídu *ProductOrService* tedy GoodRelations není sémanticky koherentní, na což je takto její uživatel upozorněn.

- Jazyk PURO byl rovněž aplikován na *analýzu a syntézu strukturních návrhových vzorů* obsažených v katalozích vzorů. Hlavní výsledky analýzy byly publikovány ve sborníku konference K-CAP 2013 [28] vydaném ACM SIGART. Jedná se zejména o zjištění, že rodina pěti strukturních vzorů navržená W3C pro řešení problému "třídy jako hodnoty vlastnosti" [18] implicitně obsahuje vzory odpovídající třem různým modelovacím problémům: dva vzory primárně zachycují vztah ke třídě jako takové (intenzi třídy), jeden zachycuje vztah k množině jejích instancí (extenzi třídy), a dva zachycují vztah k "tématu", které sice s třídou úzce souvisí, ale není totožné ani s její intenzí ani s extenzí. Dále byly zformulovány dva další vzory, které rovněž umožňují vyjádřit vztah k instancím a mají ve srovnání se stávajícím vzorem určité komparativní výhody.

[42] To, že je model PURO netriviálním protějškem ontologie OWL, kterou anotuje, bylo přitom prokázáno skutečností, že jen v 63% případů odpovídal třídě z OWL $\mathcal{B}$-typ 1. řádu; v ostatních případech šlo zpravidla o $\mathcal{B}$-typy vyšších řádů nebo o $\mathcal{B}$-vztahy.

[43] Formalizaci soustavy omezení z převážné části provedli kolegové z UK Bratislava.

[44] `http://www.heppnetz.de/projects/goodrelations/`

- Vedle detailní analýzy jednotlivých ontologií a strukturních vzorů lze PURO rovněž využít pro přibližnou vizuální analýzu tzv. *lokálního pokrytí* určité domény větším počtem ontologií OWL [6]. Na rozdíl od předchozích scénářů je zde model PURO chápán primárně na úrovni modelu *příkladu* složeného z $\mathcal{B}$-objektů a $\mathcal{B}$-vztahů (tj. jednotlivin), na které teprve navazuje jejich typování: příklad tedy může být budován z konkrétních dat, která bychom chtěli s pomocí analyzovaných ontologií zachytit. Tento pohled je intuitivní pro vizualizaci určenou k získání přehledu o způsobu modelování dat (např. oproti vizualizaci s instancemi přístupnými jen přes jejich třídy, která je používána jako výchozí ve většině ontologických editorů). Takto pojatý model PURO totiž zpravidla tvoří jediný souvislý graf. Pro jeho konstrukci M. Dudáš vyvinul a na naznačenou úlohu aplikoval vizuální editor *PURO Modeler*.[45] Uživatel nejprve vytvoří (případně načte existující) model PURO, a následně v něm vyznačuje, které z jeho prvků jsou pokryty kterou z analyzovaných ontologií. Nástroj průběžně zajišťuje vizualizaci pokrytí jednotlivými ontologiemi formou modifikovaných Vennových diagramů. Vytvořený diagram je přehledným podkladem pro rozhodování o *přepoužití* určité (ideálně, počtem minimální) množiny ontologií, která bude pokrývat všechny nebo co největší počet pojmů obsažených ve vystavovaných datech.

- Posledním detailně rozpracovaným scénářem je podpora *tvorby nových ontologií OWL* na základě modelů PURO, která je hlavním tématem rozpracované disertační práce M. Dudáše (pod vedením autora). Opírá se jak o nástroj PURO Modeler zmíněný v předchozím bodě, tak o návazný nástroj OBOWLMorph[46] sloužící k vytváření fragmentů ontologií OWL v různých stylových variantách. Aktuální verze nástroje přímo umožňuje nejen generování nových entit, ale i *přepoužití* entit z existujících ontologií, identifikovaných pomocí automatického *mapování ontologií*. Vygenerovaný fragment je následně dotvořen již v OWL, např. v editoru Protégé. Soubor vyvinutých nástrojů byl pro tuto úlohu (na dvou vzorových zadáních tvorby jednoduché ontologie) testován skupinou 26 studentů v porovnání s konvenčním přístupem, kdy je ontologie vytvářena od počátku v OWL v nástroji Protégé. Nový postup založený na nástrojích PURO dosáhl výrazně vyššího System Usability Score (SUS), přičemž ontologie vzniklé prvotně v Protégé měly lepší pokrytí pojmů ze zadání (zejména díky možnosti přímo modelovat n-ární relaci). Scénář, jeho softwarová podpora a evaluace jsou popsány v rukopisu, který je aktuálně v recenzním řízení v *J. Web Semantics*.

- Jen doplňkově (jde o slibný, ale dosud nerealizovaný scénář) ještě můžeme zmínit úlohu extrakce tzv. *minimálních uzavřených popisů* (CBD - "concise bounded descriptions") entit v RDF. Jedná se o to, že servery určené pro vystavování propojených dat v RDF by měly v reakci na zadání IRI určité entity (např. osoby, firmy nebo lokality) v požadavku HTTP vrátit ucelený soubor informací o ní. Obvykle se jedná o trojice, ve kterých se tato entita přímo vyskytuje. Pokud by však entita na druhém konci trojice byla pomocí PURO anotována jako $\mathcal{B}$-vztah (šlo by např. o instanci třídy *Manzelstvi* nebo *Prodej*), vrácený soubor informací by měl obsahovat ještě *další úroveň* trojic za touto entitou, aby nedošlo k tomu, že se příjemce pouze dozví, že je dotazovaná entita *s někým/něčím* (neupřesněným) ve vztahu manželství resp. prodeje.

4.4 Kategorizační síla jako kritérium přepoužití ontologií

Zatímco předchozí metody se primárně zaměřovaly na překlenutí strukturní heterogenity v případě ontologií, které již byly pro účely přepoužití vybrány, zde budeme předpokládat, že máme k dispozici pouze rozsáhlou výchozí kolekci ontologií a potřebujeme zhodnotit, nakolik je která z ontologií pro přepoužití v daném případě (např. pro publikování daného datasetu)

[45] `http://lod2-dev.vse.cz/puromodeler-v2/`
[46] `http://lod2-dev.vse.cz/puromodeler-v2/OBOWLMorph/`

vhodná. Předpokládáme přitom, že nás zajímá, v jakém rozsahu nám dotyčná ontologie poskytne prostředky pro jemnější *kategorizaci* takových entit, pro které už je výchozí kategorie (tj. určitá, relativně obecná pojmenovaná třída) známa. Tuto výchozí kategorii FC označíme jako *fokusovou třídu*.

Celou ontologii pak můžeme posuzovat podle toho, kolik možných speciálnějších kategorií nabízí pro FC (v případě více fokusových tříd lze tutéž analýzu udělat pro každou separátně a výsledky pak agregovat). Problémem ovšem je, že možných kategorií jako konceptových výrazů syntakticky utvořitelných ze signatury (tj. množiny všech entit) ontologie může být pro libovolnou FC mnoho (zpravidla nekonečně). Mnohé z nich přitom nebudou relevantní, např. proto, že budou spojovat neslučitelné entity a množina jejich instancí bude vždy prázdná, nebo budou naopak logicky ekvivalentní třídě FC samotné a nebudou mít tedy pro kategorizaci žádnou přidanou hodnotu.

Lze předpokládat, že *pojmenované podtřídy* FC, ať už přímé nebo nepřímé, relevantními kategoriemi budou. Jejich množina tedy tvoří jakousi dolní aproximaci množiny všech relevantních kategorií, které budeme při výpočtu fokusované kategorizační síly ontologie vzhledem k její fokusové třídě F brát do úvahy s plnou vahou. Vedle toho nás ale budou zajímat i další, již *složené* konceptové výrazy, a to zpravidla s nižšími váhovými koeficienty. Pokud tedy budeme uvažovat množinu konceptových výrazů vymezených formálním jazykem $\mathcal{L}$ a množinu vzorů používaných pro jejich detekci $P = \{p_1, ..., p_n\}$, můžeme odhad fokusované kategorizační síly ontologie O vzhledem k fokusové třídě FC vyjádřit jako

$$\widehat{FOCP}(FC, O, \mathcal{L}, P) = Occ(p_1, FC, O) * w_1 + ... + Occ(p_n, FC, O) * w_n$$

kde $Occ(p_i, FC, O)$ je funkce vracející počet výskytů vzoru p_i v O, a w_i jsou váhové koeficienty.

Pro stanovení adekvátních váhových koeficientů se nabízejí dva hlavní zdroje: zpětná vazba od uživatelů, a automatická empirická analýza – jednak samotných ontologií, jednak datových sad, které se na ni odkazují. V první fázi výzkumu byla jako zdroj využita zejména *zpětná vazba od uživatelů* k jednotlivým (složeným) konceptových výrazům různých typů. Rozlišovány byly tři typy výrazů odpovídající velmi jednoduchému jazyku $\mathcal{L}$, které lze vyjádřit variantami existenční restrikce: $FC \sqcap \exists R.C$ (tj. s hodnotou vlastnosti vymezenou pomocí pojmenované třídy), $FC \sqcap \exists R.\{i\}$ (tj. "value restriction" s hodnotou vyjádřenou konkrétním individuálem), a $FC \sqcap \exists R.\top$ (tj. s hodnotou vlastnosti "vymezenou" univerzálním "super-konceptem", tedy nijak neomezenou). Uživatelé měli za úkol rozhodnout, zde daný konceptový výraz považují za *přepoužitelnou*[47] kategorii vzhledem k určité jeho logické nadtřídě FC, nebo ne. Příklady konceptových výrazů (jednotlivých výše uvedených typů), které byly uživateli, konkrétně, 27 studenty dvou předmětů zaměřených na ontologické inženýrství resp. propojená data, relativně často vnímány jako přepoužitelné kategorie, jsou:[48]

- $Place \sqcap \exists\ isEquippedBy.AudiovisualEquipment$
- $FridgeFreezer \sqcap \exists\ styleOfUnit.\{SingleDoor\}$
- $ProgramCommitteeMember \sqcap \exists\ writeReview.\top$

V prvním případě je kategorie "místa" upřesněna pomocí třídy *AudiovisualEquipment* omezující hodnoty vlastnosti *isEquippedBy*. Ve druhém případě je sice *SingleDoor* formálně individuem, avšak v realitě jde opět o vyjádření obecné kategorie (stylu chladničky) – zde vidíme spojitost s modely PURO, kde by toto individuum bylo modelováno jako $\mathcal{B}$-typ. Ve

[47] Pro účel takového rozhodování bylo neformálně vymezeno, že za přepoužitelnou kategorii bude uživatel považovat takovou, u které by ho "nepřekvapilo", kdyby byla vyjádřena i jako pojmenovaná třída. Kategorie, které takto vnímá většina uživatelů ("ontologistů") pak označíme jako *ontologistické kategorie*.

[48] Prvním členem konjunkce je vždy fokusová třída, kterou výraz specializuje. Jmenné prostory ontologií pro stručnost neuvádíme.

třetím případě je vymezení kategorie dáno pouze vlastností *writeReview*; omezující kategorie hodnoty vlastnosti ("recenze") je zde přítomna implicitně v jejím názvu, proto kategorie dává smysl i bez explicitního upřesnění hodnoty ve struktuře výrazu.

Z četnosti odpovědí uživatelů (vyjádřených na Likertově škále) byla pracovně odvozena empirická pravděpodobnost, že náhodně vybraný konceptový výraz daného typu bude "průměrným uživatelem" chápán jako přepoužitelná kategorie; tato pravděpodobnost může být použita jako váhový koeficient ve výše uvedeném vzorci pro $\widehat{FOCP}$. Nejvyšší hodnoty (cca 0,7) dosáhl vzor $FC \sqcap \exists R.\{i\}$, následován $FC \sqcap \exists R.C$ (cca 0,5); nejnižší, ale stále nezanedbatelná pravděpodobnost (cca 0,3) pak byla zjištěna u vzoru $FC \sqcap \exists R.\top$.

V ontologiích se ovšem nevyskytují samotné konceptové výrazy, ale logická tvrzení. Na ně je nutno aplikovat ontologické vzory rovněž uvedené ve vzorci pro $\widehat{FOCP}$, abychom získali hodnoty jednotlivých $Occ(p_i, FC, O)$. Mírně zjednodušeným příkladem takového "konstrukčního" vzoru pro výraz $FC \sqcap \exists R.C$ je

$$\exists D \ \ (\ \ R \ \texttt{rdfs:domain} \ FC \ \ \wedge \ \ R \ \texttt{rdfs:range} \ D \ \ \wedge \ \ C \ \texttt{rdfs:subClassOf} \ D \ )$$

tj. výraz zkonstruujeme tehdy, jestliže má vlastnost R jako svůj definiční obor (ať už přímo, nebo s využitím dědičnosti) fokusovou třídu FC, a zároveň má jako svůj obor hodnot určitou třídu D takovou, že třída C je její podtřídou.

Téma fokusované kategorizační síly bylo dosud in extenso (včetně výsledku automaticky zpracovaných analýz výskytu relevantních vzorů nad rozsáhlými kolekcemi ontologií, které provedl O. Zamazal) zpracováno v jediné publikaci, příspěvku na evropské konferenci EKAW 2016 [33]. Ten byl (v konkurenci 51 prezentovaných příspěvků vybraných ze 171 podaných) zařazen mezi 5 příspěvků nominovaných na Cenu pro nejlepší příspěvek,[49] což spolu s velmi pozitivními verbálními ohlasy indikuje aktuálnost tohoto směru výzkumu.

5 Kontext výzkumu: pracoviště, tým a projekty

Výzkum autora prezentovaný v tomto textu je neodmyslitelně spojen s týmovými vědeckými (i pedagogickými) aktivitami katedry a užší pracovní skupiny. Tyto aspekty, včetně grantové podpory výzkumu, zde proto alespoň stručně shrnujeme.

5.1 Zaměření pracoviště a vědecký tým

Katedra informačního a znalostního inženýrství se již od svého vzniku v r. 1990 intenzivně zabývala problematikou umělé inteligence a znalostního inženýrství (např. reprezentace znalostí v pravidlových expertních systémech). Ve druhé polovině 90. let k tomu přistoupil i zájem o nastupující webové technologie, opět zejména ve spojení s inteligentními systémy a reprezentací znalostí. Autor patřil k pracovníkům, kteří se na oba okruhy témat již v té době zaměřovali, a zapojení do mezinárodní komunity sémantického webu (a ontologického inženýrství) pro něj bylo přirozeným pokračováním tohoto směru. Již krátce po r. 2000 začalo pod jeho vedením krystalizovat jádro neformální pracovní skupiny pro sémantický web, ačkoliv k jejímu explicitnímu vymezení došlo až v r. 2015 v rámci strukturování katedry jako celku do čtyř pracovních skupin. V současnosti skupina *Semantic Web and Ontological Engineering* (SWOE, viz webová stránka `http://kizi.vse.cz/swoe`) zahrnuje tři stálé pracovníky katedry, čtyři doktorandy, a navíc několik externích spolupracovníků a studentů magisterského stupně studia. Hlavní dva pilíře činnosti skupiny představují *ontologické inženýrství založené na vzorech* (přibližně jde o tématiku odpovídající profesorské přednášce) a *tvorba a zpracování propojených dat* – převážně, ač ne výlučně, v oblasti veřejné správy.

[49] `http://ekaw2016.cs.unibo.it/?q=awards`

Pokud jde o členy skupiny jmenovitě, výzkum autora v oblastech relevantních profesorské přednášce se v největší míře opírá o spolupráci s *Ondřejem Zamazalem* (roz. Švábem). Spolupráce trvá již od r. 2004 (kdy byl Ondřej ještě studentem bakalářského studia) až do současnosti (kdy jako odborný asistent již sepisuje svou habilitační práci). Lze říci, že výzkum aplikace ontologických vzorů v oblasti mapování ontologií (kolekce OntoFarm a aktivity kolem ní), a do značné míry i v oblasti transformace ontologií (projekt PatOMat), sice autor inicioval, ale O. Zamazal na něm postupně převzal většinový podíl, především pokud jde o implementační a experimentální činnosti, ale i zahraniční spolupráci. Dalším dlouhodobě spolupracujícím kolegou je *Miroslav Vacura*, který se ontologickému inženýrství rovněž věnuje od poloviny první dekády, kdy se zapojil do projektu EU K-Space a spoluvyvinul v něm vysoce uznávanou Core Ontology of Multimedia (COMM). Na výzkumu prezentovaném v tomto textu se podílel zejména účastí na návrhu formalismu PURO. Z mladších pracovníků (doktorandů) je pro oblast ontologického inženýrství klíčové zapojení *Marka Dudáše*, který vyvinul softwarové komponenty pro interaktivně řízené transformace ontologií (GUIPOT), tvorbu transformačních vzorů (TPE), pro tvorbu modelů PURO a jejich transformaci do jazyka OWL (PURO Modeler, OBOWLMorph), a pro tvorbu a vizualizaci ontologických souhrnů datových sad (LODSight). Hodná zmínky je účast dalšího doktoranda *Jindřicha Mynarze*, který je především expertem na propojená data včetně tvorby jejich slovníků, a dále studentů *Simone Serry* a *Tomáše Hanzala*, kteří přispěli implementacemi a experimenty pro modely PURO. Ve stadiu recenzního řízení je společný článek autora s "lingvistickým expertem" skupiny *Petrem Strossou*, zaměřený na lexikální analýzu ontologií.

Poznamenejme, že mezi výzkumné výsledky prezentované ve významnějším rozsahu v tomto textu (sekce 4.2, 4.3 a 4.4) byly zařazeny takové, u kterých je podíl vlastního výzkumu autora zásadní (což je reflektováno i prvoautorstvím souvisejících publikací).

5.2 Řešené grantové projekty

Popsaný výzkum byl podporován mezinárodními, národními i fakultními grantovými projekty. Uvádíme stručný přehled těch nejrelevantnějších, s upřesněním výzkumných výsledků (spadajících pod tématiku profesorské přednášky) v jejich rámci dosažených týmem autora:

Projekty EU:

- IST FP6-507482, *Knowledge Web* (místní řešitel V. Svátek): kolekce OntoFarm a její využívání pro testování mapovacích systémů
- ICT FP7-257943, *Creating Knowledge out of Interlinked Data* (LOD2, místní řešitel V. Svátek): transformace lexikálního aspektu v rámci revize ontologií.

Projekty financované na národní úrovni:

- GA ČR P202/10/1825 "PatOMat – Automation of Ontology Pattern Detection and Exploitation" (řešitel V. Svátek): návrh, implementace a využití metody transformace ontologií založené na transformačních vzorech
- Mobilitní česko-slovenský projekt MŠMT 7AMB12SK020, "Logical Aspects of Adaptable Ontological Schemas" (LAAOS, řešitel za českou stranu V. Svátek): návrh formalismu PURO a jeho využití pro testování koherence ontologie a analýzu strukturních návrhových vzorů.

Projekty IGA VŠE:

- 12/06 "Integration of approaches to ontological engineering: design patterns, mapping and mining" (řešitel O. Šváb)

- 20/07 “Combination and comparison of ontology mapping methods and systems” (řešitel O. Šváb)
- F4/34/2014 “Vizualizace dat sémantického webu s využitím struktury ontologických schémat” (řešitel M. Dudáš)
- F4/90/2015 “Generování stylových variant sémantických datových schémat s využitím generalizované vizualizace dat” (řešitel M. Dudáš)
- F4/28/2016 “Využití automatického mapování a lingvistické transformace při generování sémantických datových slovníků” (řešitel M. Dudáš).

6 Závěr

Přestože prezentovaný výzkum zahrnuje různorodé metody, společným jmenovatelem podstatné části z nich je

- cílová úloha – efektivní *přepoužití* entit z existujících ontologií v novém kontextu
- zaměření na problém *strukturní heterogenity* ontologií, který je pro přepoužívání entit častou překážkou
- využití *ontologických vzorů*, ať už spíše empiricky objevených nebo systematicky navržených.

Vzhledem k vnímání přepoužívání entit jako “nejlepší praxe” lze očekávat, že alespoň některé z navržených metod si najdou cestu do praxe.

Nejpřímočařejší cestu v tomto směru může mít výpočet *fokusované kategorizační síly*, jelikož, v případě předchozího vyladění váhových koeficientů pro různé typy konceptových výrazů a vzorů, nevyžaduje žádné dodatečné úsilí od uživatele: jeho výstupem bude numerická váha, kterou lze snadno integrovat s dalšími měrami vhodnosti pro přepoužití, např. založenými na *popularitě* entit/ontologií nebo na *důvěryhodnosti* tvůrce ontologie [19].

Vysoký potenciál má i tvorba ontologie na základě *modelů PURO*; zde je ale limitujícím faktorem nutnost udržovat uživatelsky přívětivý software. Bez podpory takového software totiž nevzniknou modely PURO v kvantitě a kvalitě dostatečné na to, aby se celé “nové paradigma” v praxi prosadilo a dalším uživatelům se vyplatilo se modelování v PURO naučit.

Transformační technologie z projektu PatOMat si již získala určitou odezvu zejména v kruzích biomedicínské informatiky, kde by mohla pomoci řešit problém vzájemné adaptace nezávisle vzniklých a následně integrovaných modulů ontologií. Toto její využití ovšem leží zčásti mimo oblast otevřeného (sémantického) webu a propojených dat, která je pro autora a jeho pracovní skupinu v tuto chvíli prioritní. Vedle toho se nabízí možnost využít ji i v kombinaci s konceptem fokusované kategorizace: konceptové výrazy predikované jako ontologistické kategorie pro určitou fokusovou třídu mohou být automaticky transformovány na pojmenované třídy, které mohou být uživateli doporučeny pro dodatečné zařazení do ontologie.

Od začátku prezentovaného výzkumu se oborové zaměření autora a jeho skupiny také stále více posouvá od širokého spektra aplikací k dominanci oblastí, pro které jsou na půdě VŠE zvláště vhodné podmínky: zpracování dat z oblasti (zejména, elektronického) obchodu a veřejné správy. I v této oblasti se pro ontologické inženýrství stále objevují nové výzvy, na které výzkum skupiny bude reagovat.

Reference

[1] *Resource Description Framework (RDF)*. Webová stránka konsorcia W3C, online `http://www.w3.org/RDF/`.

[2] *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)*. W3C Recommendation, 11 December 2012, online `https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/`.

[3] Baader F. et al.: The description logic handbook: theory, implementation, and applications. Cambridge University Press New York, NY, USA, 2003.

[4] Blomqvist E., Hammar K., Presutti V.: Engineering Ontologies with Patterns – The eXtreme Design Methodology. In: Ontology Engineering with Ontology Design Patterns, IOS Press, 2016: 23–50.

[5] Dudáš M., Svátek V., Mynarz J.: Dataset Summary Visualization with LODSight. In: ESWC (Satellite Events), LNCS 9341, Springer, 2015: 36–40.

[6] Dudáš M., Hanzal T., Svátek V.: What Can the Ontology Describe? Visualizing Local Coverage in PURO Modeler. VISUAL@EKAW 2014, CEUR Workshop Proceedings 1299, CEUR-WS.org 2014: 28–33.

[7] Euzenat J., Shvaiko P.: *Ontology Matching*, Second Edition. Springer 2013, ISBN 978-3-642-38720-3.

[8] Gangemi A.: Ontology Design Patterns for Semantic Web Content. International Semantic Web Conference, LNCS 3729, Springer, 2005: 262–276.

[9] Gómez-Pérez A., Fernández-López M., Corcho O.: Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. Advanced Information and Knowledge Processing, Springer 2004, ISBN 978-1-85233-551-9, pp. 1–362.

[10] Gruber T.R.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2) (1993).

[11] Guarino N., Welty C. A.: An Overview of OntoClean. In: Handbook on Ontologies, Springer, 2009: 201–220.

[12] Heath T., Bizer C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers 2011.

[13] Hitzler P., Gangemi A., Janowicz K., Krisnadhi A., Presutti V.: *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*. Studies on the Semantic Web 25, IOS Press 2016, ISBN 978-1-61499-675-0

[14] Lawrynowicz A., Potoniec J., Robaczyk M., Tudorache T.: Discovery of Emerging Design Patterns in Ontologies Using Tree Mining. Přijato pro: *Semantic Web*, online `http://www.semantic-web-journal.net/content/discovery-emerging-design-patterns-ontologies-using-tree-mining-0`.

[15] Meilicke C., Garcia-Castro R., Freitas F., van Hage W. R., Montiel-Ponsoda E., Ribeiro de Azevedo R., Stuckenschmidt H., Šváb-Zamazal O., Svátek V., Tamilin A., Trojahn dos Santos C., Wang S.: MultiFarm: A benchmark for multilingual ontology matching. *J. Web Sem.* 15: 62–68 (2012).

[16] Mikroyannidi E., Iannone L., Stevens R., Rector A. L.: Inspecting Regularities in Ontology Design Using Clustering. In: International Semantic Web Conference (1) , LNCS 7031, Springer, 2011: 438–453.

[17] Nečaský, M., Klímek, J., Mynarz, J., Knap, T., Svátek, V., Stárka, J.: Linked data support for filing public contracts. *Computers in Industry* 65(5), 862–877 (2014).

[18] Noy, N. (ed.): *Representing Classes As Property Values on the Semantic Web.* W3C Working Group Note 5 April 2005, online `http://www.w3.org/TR/swbp-classes-as-values/`.

[19] Stavrakantonakis I., Fensel A., Fensel D.: Linked Open Vocabulary Ranking and Terms Discovery. In: SEMANTiCS 2016, ACM, 2016: 1–8.

[20] Schaible J., Gottron T., Scherp A.: Survey on Common Strategies of Vocabulary Reuse in Linked Open Data Modeling. In: ESWC 2014, LNCS 8465, Springer 2014: 457–472.

[21] Scharffe F., Ding Y., Fensel D.: Towards Correspondence Patterns for Ontology Mediation. In: OM 2007, CEUR Workshop Proceedings 304, CEUR-WS.org 2008.

[22] Scharffe F., Zamazal O., Fensel D.: Ontology alignment design patterns. *Knowl. Inf. Syst.* 40(1): 1–28 (2014).

[23] Serra S.: Background annotation of entities in Linked Data vocabularies. Diplomová práce, KIZI FIS, 2013.

[24] Staab S., Studer R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, Springer 2009, ISBN 978-3-540-70999-2.

[25] Svátek V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: Business Information Systems – BIS 2004. Poznan, Wydawnictwo Akademii Ekonomicznej w Poznaniu, 2004, 437—446. ISBN 83-7417-019-0.

[26] Svátek V., Dudáš M., Zamazal O.: Adapting ontologies to best-practice artifacts using transformation patterns: Method, implementation and use cases. *J. Web Sem.* 40: 52–64 (2016).

[27] Svátek V., Homola M., Kluka J., Vacura M.: Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models. In: OWLED 2013, CEUR Workshop Proceedings 1080, CEUR-WS.org 2013.

[28] Svátek V., Homola M., Kluka J., Vacura M.: Mapping structural design patterns in OWL to ontological background models. In: K-CAP 2013, ACM, 2013: 117–120.

[29] Svátek V., Serra S., Vacura M., Homola M., Kluka J.: B-Annot: Supplying Background Model Annotations for Ontology Coherence Testing. In: WoDOOM 2014, CEUR Workshop Proceedings 1162, CEUR-WS.org 2014: 59–66.

[30] Svátek V., Šváb-Zamazal O., Presutti V.: Ontology Naming Pattern Sauce for (Human and Computer) Gourmets. In: WOP 2009, CEUR Workshop Proceedings 516, CEUR-WS.org 2009.

[31] Svátek V., Vacura M.: Ontologické inženýrství na sémantickém webu. In: Umělá inteligence (6). Praha : Academia, 2013, s. 149-–168. 490 s. ISBN 978-80-200-2276-9.

[32] Svátek V., Zamazal O., Dudáš M.: Using ODPs for Ontology Transformation. In: Ontology Engineering with Ontology Design Patterns, IOS Press, 2016: 245–266.

[33] Svátek V., Zamazal O., Vacura M.: Categorization Power of Ontologies with Respect to Focus Classes. In: EKAW 2016, LNCS 10024, Springer, 2016: 636–650.

[34] Šváb O., Svátek V., Berka P., Rak D., Tomášek P.: Ontofarm: Towards an experimental collection of parallel ontologies. In: Poster Track of ISWC, 2005.

[35] Šváb O., Dudáš M., Svátek V.: User-Friendly Pattern-Based Transformation of OWL Ontologies. In: EKAW 2012, LNCS 7603, Springer, 2012: 426–429.

[36] Šváb O., Svátek V., Stuckenschmidt H.: A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In: ESWC 2007, LNCS 4519, Springer, 2007: 655–669.

[37] Šváb-Zamazal O., Schlicht A., Stuckenschmidt H., Svátek V.: Constructs Replacing and Complexity Downgrading via a Generic OWL Ontology Transformation Framework. In: SOFSEM 2013, LNCS 7741, Springer, 2013: 528–539.

[38] Šváb-Zamazal O., Svátek V.: Analysing Ontological Structures through Name Pattern Tracking. In: EKAW 2008, LNCS 5268, Springer, 2008: 213–228.

[39] Šváb-Zamazal O., Svátek V., Iannone L.: Pattern-Based Ontology Transformation Service Exploiting OPPL and OWL-API. In: EKAW 2010, LNCS 6317, Springer, 2010: 105–119.

[40] Zamazal O., Bühmann L., Svátek V.: Checking and repairing ontological naming patterns using ORE and PatOMat. In: WoDOOM 2013, CEUR Workshop Proceedings 999, CEUR-WS.org 2013, 69–76.

[41] Zamazal O., Svátek V.: PatOMat - Versatile Framework for Pattern-Based Ontology Transformation. *Computing and Informatics* 34(2): 305–336 (2015).

[42] Zamazal O., Svátek V.: The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *J. Web Sem.*, in Press, 2017.

[43] Zamazal O., Svátek V., Scharffe F., David J.: Detection and Transformation of Ontology Patterns. In: Knowledge Discovery, Knowledge Engineering and Knowledge Management. Berlin: Springer, 2011, 210—223. ISBN 978-3-642-19031-5. ISSN 1865-0929.

Abstract

Ontological Model Reuse on the Semantic Web. The reuse of ontologies on the semantic web, expressed using the OWL language, is hindered by their structural heterogeneity, which is caused by the richness of this language and by pragmatic concerns of the ontology designers. Four different ontology reuse scenarios are formulated and corresponding methods for bridging the structural heterogeneity described, referring to previous research of the author. The cases (and methods) correspond, in turn, to: ontology alignment (considering complex, 'heterogeneous' correspondences); best-practice ontology or pattern inclusion into a legacy ontology (requiring structural adaptation of the legacy ontology); multiple ontology analysis and synthesis (in different structural variants) from ontological background models; reuse of ontologies offering high focused categorization power (based not only on the number of explicit subclasses of the focus class but also on compound OWL concept expressions). The institutional and funding context of the research is also briefly summarized.