

Ontologické inženýrství

Vojtěch SVÁTEK¹, Miroslav VACURA²

¹*Katedra informačního a znalostního inženýrství, Vysoká škola ekonomická v Praze
nám. W. Churchilla 4, 130 67 Praha 3
svatek@vse.cz*

²*Katedra filosofie, Vysoká škola ekonomická v Praze
nám. W. Churchilla 4, 130 67 Praha 3
vacuram@vse.cz*

Abstrakt. Ontologické inženýrství je moderní oblastí informatiky, která se zaměřuje na návrh, implementaci a aplikaci ontologií: sdílených a (převážně) formalizovaných znalostních modelů podporujících korektní modelování reality, sémantickou interoperabilitu a automatické odvozování v softwarových aplikacích. Ontologické inženýrství se původně vyvíjelo v rámci umělé inteligence, postupně se však posouvalo k praktickému využití při návrhu aplikací a zpřístupňování datových zdrojů, a v současné době hraje ústřední roli v iniciativě tzv. sémantického webu. Moderní pojetí ontologie v informatice se v řadě bodů liší od klasické ontologie jako součásti filozofie, ovšem má s ní také významné styčné body, kterými jsou zejména tzv. základní ontologie zachycující nejobecnější kategorizaci entit. Praktická tvorba ontologií se v současnosti opírá o sofistikované metodiky, zahrnující i využití základních ontologií a z nich vycházejících tzv. obsahových návrhových vzorů. Předností dnes nejrozšířenějšího jazyka pro reprezentaci ontologií, OWL, je existence mapování na rodinu tzv. deskripčních logik, které umožňuje na navržené ontologie aplikovat odvozovací nástroje. Vazba na exaktní logickou interpretaci ovšem na druhé straně způsobuje, že konkrétní implementace dané konceptualizace není triviální. Hlavním prostředkem pro přizpůsobení konceptualizace danému reprezentačnímu jazyku a logickému kalkulu jsou tzv. logické návrhové vzory. Implementace a využívání ontologií se ve finální fázi uskutečňuje prostřednictvím specializovaných softwarových prostředků; jedná se především o editory (z nichž nejznámější je Protégé), zpravidla vybavené zásuvnými moduly pro řadu dalších úloh s ontologiemi souvisejících.

Klíčová slova: ontologie, deskripční logika, metodika, návrhové vzory.

1 Úvod

V posledních letech se tématika ontologií a ontologického inženýrství diskutuje v kruzích teoretické i aplikované informatiky stále častěji. Mnozí si kladou otázku: jedná se skutečně o perspektivní technologii, která v příštích desetiletích významně ovlivní koncepci informačních systémů, nebo jde spíše o módní vlnu, která za několik let beze stop odezní? V tomto stručném materiálu neposkytneme na uvedenou otázku přímočarou odpověď. Pokusíme se však, pro čtenáře, který je ochoten připustit

Ontologické inženýrství

variantu první, vymezit určité mantinely, co je možné a vhodné za ontologie pokládat, vysvětlit, z jakých kořenů jejich výzkum v informatice pochází, a naznačit, jaký okruh metod a nástrojů lze při jejich tvorbě a další práci s nimi využít. Text vychází z velkého množství podkladových materiálů i z vlastních zkušeností autorů. Snahou přitom bylo, vedle podrobněji rozebraných dílčích problémů, zejména upozornit na užitečné zdroje, které jsou dostupné na webu nebo případně v knižní podobě¹.

Pojem ontologie má za sebou několik století pozvolného vývoje v kontextu filosofie, na který vcelku plynule navázal (původně širší informatické veřejnosti nepříliš známý) výzkum možností ontologií jako prostředku formální reprezentace znalostí v kontextu umělé inteligence. Výrazný nárůst popularity „informatických“ ontologií (a také termínu samotného, i ve spíše okrajových významech) však vyplynul až z jejich sepětí s prostředím WWW na konci 90. let, ztělesněné zejména širokou iniciativou *sémantického webu*. Text se bude zabývat ontologiemi jak v obecnějším smyslu, tak i problémy spojenými s jejich reprezentací a logickým odvozováním nad nimi v rámci konkrétního formalismu – jednoznačně nejrozšířenějšího „webového“ ontologického jazyka OWL podloženého tzv. deskripční logikou. Vzhledem k omezenému prostoru se však jen lehce dotkneme problematiky syntaktické reprezentace, adresování a dalších otázek spojených s bezprostředním využíváním ontologií na WWW², a soustředíme se na konceptuální a logické aspekty ontologického inženýrství.

Zmíněné důrazy se promítají do celkové struktury textu. Kapitola 2 se pokouší propojit ontologii jako tradiční *filosofickou disciplínu* se soudobým výzkumem ontologií v informatice. Kapitola 3 podává obecnou *typologii* ontologií, a podrobněji se věnuje dvěma hlavním typům ontologií z obsahového hlediska: *základním* ontologiím – včetně jejich tradičního filosofického zázemí – a zpravidla pragmatičtější pojatým *doménovým* ontologiím. Kapitola 4, věnovaná stručnému přehledu *reprezentačních jazyků* pro ontologie, je do jisté míry odbočkou ve výkladu. Po ní ovšem následuje kapitola 5, která je opět zaměřena na *věcný obsah* ontologií, přesně řečeno na prostředky podporující jeho korektní a efektivní tvorbu. Kapitola 6 se pak věnuje (specificky pro jazyk OWL a jemu podobné) vztahu mezi ontologiemi a *logickým* odvozováním, jak z hlediska odvozovacích procedur jako takových, tak z hlediska potřeby zohlednit logickou reprezentaci a odvozování při samotné tvorbě ontologie. Následují dvě kapitoly zaměřené spíše na praxi: kapitola 7 poskytuje rychlý přehled vybraných *softwarových nástrojů* pro práci s ontologiemi, a kapitola 8 referuje o některých příkladech *aplikace* ontologií v reálném prostředí. Závěrečná kapitola 9 pak už jen rekapituluje poslání textu.

¹ Zde je na místě zmínit se o přehledové knize o ontologickém inženýrství zkušených autorů z Universidad Politécnica de Madrid, která již po třech letech od prvního vydání letos vychází ve vydání druhém [13].

² Budeme předpokládat, že zejména databázově orientovaný čtenář má již o problematice *sémantického webu* jako takového určité povědomí. Problematikou ontologií v kontextu WWW se mj. zabýval předchozí tutoriál jednoho z autorů [51]. Současný tutoriál věnuje podstatně menší prostor např. koncepci *sémantického webu* nebo historii ontologických jazyků, naopak podrobněji rozebírá tvorbu ontologického obsahu či deskripční logiku.

2 Filosofické kořeny

Mezi první ontologické koncepce patřily už názory předsockratických myslitelů,³ které se povětšinou omezily na snahu redukovat rozličnost smyslového světa na určitý základní princip (arché). Významnější pokrok přineslo až myšlení Platóna [37] a posléze jeho žáka Aristotela, u kterého se zastavme, protože u toho nacházíme první ontologickou koncepci, kterou můžeme považovat za předchůdce moderních ontologií. Aristotelés svoji teorii předkládá ve spisech *Metafyzika* a *Kategorie*, kde mluví o tzv. *první vědě*, která kromě ostatních určení zkoumá zejména „jsoucí jakožto jsoucí a to, co mu samému náleží“. Výrazy „metafyzika“, „ontologie“ a aristoteléska „první filosofie“ se často ve filosofii používají jako synonyma.

Zatímco se tedy vědy speciální věnují pouze určitému vymezenému okruhu jsoucna nebo určitým vymezeným aspektům skutečnosti, ontologie se v tomto pojetí věnuje tomu, co je všemu jsoucímu společné. Aristotelés se však už nesnaží o pouhou redukci veškeré skutečnosti na jeden základní princip jako předsockratictí myslitelé, ale snaží se rozpoznat základní vlastnosti a rozčlenění veškeré skutečnosti. Základním protikladnými charakteristikami pak jsou *akt a potence*, následně pak *substance a akcident*, *látka a forma*. Pro nás podstatné je rozlišení tzv. deseti *kategorií*, tedy substance⁴ a devíti akcidentů: kvantita, kvalita, relace, kde, kdy, poloha, mít, činnost a trpnost. Těchto deset kategorií mělo vyčerpávat všechny základní třídy toho, co může být. Nezávislé bytí přitom měla vždy pouze jsoucna kategorie substance, zatímco jsoucna ostatních kategorií vždy nějak na bytí nějaké substance participovala. Například červená, jakožto určitá kvalita, nemá samostatné bytí, ale vždy existuje jen prostřednictvím nějaké červené věci. Pro nás je přitom zajímavé, že kategorie měly vnitřní hierarchickou strukturu. Není přitom jasné, nakolik sám Aristotelés tuto hierarchickou strukturu např. v případě přírodovědných oborů rozpracoval, protože ačkoli víme, že se této problematice věnoval, téměř všechny tyto spisy jsou ztraceny.

Nejstarším dochovaným rozpracováním této hierarchické struktury je tak spis novoplatonika Porfýria (234-305) nazvaný *Isagoge*, v podstatě úvod do Aristotelova spisu *Kategorie*. Toto dílo obsahuje tzv. Porfýriův strom (*Arbor porphyriana*), který je hierarchickou klasifikací aristotelových substancí, ze které dodnes do jisté míry vychází klasifikační systém používaný v biologii [23].

Alternativní klasifikační systém se objevil až ve 13. století, kdy se Ramon Llull (1232-1316) ve svém hlavním díle *Ars Magna* pokusil vypracovat logicko-kombinatorický grafický systém pro klasifikaci veškerého poznání, nezávislý na jazyku, náboženství a kultuře. Hlavní vliv tohoto díla byl na rozvoj matematické kombinatoriky např. u Leibnize, jakožto klasifikační systém se neprosadil. Ve středověku se pak rozvíjením klasifikačních přístupů, především v Porfýriově tradici, zabýval např. William Ockham (1280-1349) a další.

³ Za zakladatele filosofické ontologie je tradičně považován Parmenidés a za první dílo věnující se ontologii jeho báseň *O přírodě*.

⁴ Pro Aristotela je základní (první) substancí konkrétní objekt (např. Sókratés), následně zavádí tzv. druhé substance, což jsou obecniny (např. člověk). Substance (řec. ousia, lat. *sustantia* – to, co stojí pod) tak představuje jakousi základní jednotku skutečnosti, to skutečně jsoucí, proto u různých filosofů nabývalo toto slovo různých významů (např. pro atomisty jsou atomy substancemi).

Ontologické inženýrství

Samotný výraz „ontologie“ je poměrně nový, objevil se v 17. století v pracích německých filosofů R. Göckela [18] a J. Lorharda [24], a prosazený byl patrně Ch. Wolffem, jedním z nejvýznamnějších německých filosofů před I. Kantem, který ho použil pro nahrazení tradičního termínu „metafyzika“ ve smyslu nauky o jsoucnu a bytí jako takovém, v spise *Philosophia prima sive ontologia* (1736) [35,44]. Wolff byl systematickým filosofem, který se ve svém díle snažil integrovat aristotelskou a scholastickou ontologii s myšlenkami Descarta a Leibnize, s použitím eukleidovských axiomaticko-deduktivních principů. Pro zkoumání otázek klasifikační povahy, tj. jaké třídy jsoucnen existují a jaké jsou mezi nimi vztahy, se používá označení *formální* nebo *deskriptivní* ontologie. Ze současných myslitelů je pak v tomto ohledu zajímavý zejména P. F. Strawson, a jeho spis *Individuals* [48], který je věnován mj. právě otázkám klasifikace.

V oblasti informatiky poprvé výraz ontologie použil pravděpodobně J. McCarthy v textu o nemonotónním odvozování [28]. Do souvislosti s informatikou dával ontologie v 80. letech i J. F. Sowa [45] v kontextu tématiky modálních logik. Obvykle citovaná definice se v roce 1993 objevuje v textu T. Grubera, který definuje ontologii jako „explicitní specifikaci konceptualizace“ [15]. J. F. Sowa pak navrhuje v roce 2000 následující definici: „Předmětem ontologie je studium kategorií věcí, které existují nebo mohou existovat v určité doméně. Výsledek tohoto studia, nazývaný ‘ontologie’ je katalog věcí, jejichž existenci předpokládáme v dané doméně *D*, z perspektivy osoby používající jazyk *L*, aby mluvila o *D*.“ [46] V tomto kontextu bývá také někdy doporučováno, aby pro označení vědecké disciplíny bylo používáno výrazu *Ontologie*, zatímco pro informační artefakt, který je výsledkem analýzy určité domény (případě pro studium konkrétní domény), výrazu *ontologie* [19,35].

V oblasti tvorby ontologií přitom můžeme rozlišit dva základní přístupy: *shora dolů* („top-down“) přístup, který je typický pro filosofické nebo filosoficky inspirované ontologie, které začínají pokusem rozčlenit oblast všeho, co je, na určité základní kategorie – první příklad jsme již viděli u Aristotela. Druhým přístupem je *zdola nahoru* („bottom-up“), který vychází z důkladného popisu konkrétních objektů bezprostředního zájmu dané aplikace. V důsledku tak výsledkem jsou dva odlišné typy informačních artefaktů, v případě postupu shora dolů je výsledkem *základní ontologie*, naopak výsledkem postupu zdola nahoru⁵ bývají *doménové ontologie*. Podrobněji se rozčlenění typů ontologií a jejich významným reprezentantům věnuje následující kapitola.

3 Typy ontologií

3.1 Stručný přehled

Ontologie mohou být klasifikovány na základě některých významných vlastností [6]. První takovou vlastností je *operacionalizace*, která rozlišuje ontologie *hrubé* (coarse) a *jemné* (fine-grained). Hrubé ontologie se vyznačují použitím jazyka s malou expresivitou a malým počtem axiomů (viz část 4). Jemné ontologie naopak využívají vysoce expresivní modelovací jazyk a zahrnují obvykle velké množství axiomů.

⁵ Případně jeho modifikace „od středu ven“ („middle-out“), viz část 5.2.

Tutoriál

Jemné ontologie přitom mohou přinášet vyšší nároky na odvozování. Někdy se také jemné ontologie označují jako referenční či vývojové (development-time), zatímco hrubé jako sdílené (shared) či provozní (run-time) [19].

Další důležitou vlastností je *expresivita*⁶, na základě které dělíme ontologie na těžké (heavy-weight) a lehké (light-weight). Těžké ontologie vyjadřují ontologické závazky explicitně a obsahují rozsáhlou axiomatizaci. Lehké ontologie vyjádření ontologických závazků neobsahují a často se soustředí jen na ty oblasti, které jsou autory považovány za relevantní.

Významnou vlastností je také *specifičnost* ontologie. Ontologie může být obecná (generic), jádrová (core) a doménová (domain). Obecné ontologie se věnují konceptům, které přesahují rozsah jednotlivých domén (např. proces, událost), jádrové modelují centrální koncepty určité domény, doménové ontologie obsahují koncepty specifické pro danou doménu. Základní ontologie jsou pak obecné, ovšem ne každá obecná ontologie je základní.

Ontologické závazky (ontological commitments) jsou určité základní teoretické předpoklady týkající se přístupu k modelování reality, z nichž daná ontologie vychází, a které mohou být explicitně či implicitně vyjádřeny (někdy se pro souhrn ontologických závazků používá i výraz „cognitive bias“). Pokud používáme určitou ontologii, měli bychom si být vědomi, jaké ontologické závazky s sebou nese, protože to zároveň omezuje možnou integraci s ontologiemi nesoucími jiné ontologické závazky. V následující části vyjdeme z analýz v [26,27,40] a stručně popíšeme základní typy ontologických závazků.

Ontologie se může zahrnovat *jednotliviny*, *obecniny* nebo obojí. Jednotliviny jsou entity, které nemohou mít instance, zatímco obecniny je mít mohou (ač zrovna v daný moment žádnou mít nemusí).

Ontologie může být buď *deskriptivní* nebo *revizionistická*. Deskriptivní ontologie vychází ze běžného chápání skutečnosti (*common sense*), bývá založena na vztazích vyplývajících z běžného jazyka a obvyklého porozumění, vývoj ontologie pak vychází především ze studia jazyka. Revizionistická ontologie se na přirozený jazyk a běžné porozumění neváže, snaží se vycházet z vědeckých a odborných přístupů a v případě, že jsou v rozporu s běžným rozuměním, dává jim přednost [49].

Ontologie můžeme také rozdělit na *multiplikativní* nebo *redukcionistické*. Redukcionistický přístup se snaží vycházet z malé skupiny základních konceptů a ostatní koncepty rekonstruovat z této malé základní skupiny primitivních prvků. Multiplikativní ontologie se snaží o co největší expresivitu i za cenu větší složitosti a většího počtu základních konceptů. Multiplikativní přístup také typicky připouští kolokalizaci entit (více entit se může vyskytovat na jedné časoprostorové lokaci).

Při modelování skutečnosti můžeme volit také buď *posibilistický* nebo *aktualistický* přístup. Aktualistická ontologie popisuje pouze to, co aktuálně existuje. Posibilistická popisuje i to, co může existovat, tzv. *possibilia*.

Podobná volba je i s ohledem na čas – buď *presentistický* nebo *eternalistický* přístup. Presentistický přístup modeluje jen to, co v daný okamžik aktuálně existuje. Eternalistický přístup modeluje vše, co existovalo v minulosti, existuje nyní či bude existovat v budoucnosti. Vše to je považováno za existující.

⁶ Expresivita ontologie se nemusí odvíjet od expresivity použitého jazyka, ale týká se obsahu.

S tím související je i volba 3D nebo 4D modelovacího přístupu, která má vliv na modelování změn. Při použití třírozměrného modelovacího přístupu jsou běžné předměty plně přítomny v každém časovém okamžiku a jsou proměnné, neboli v různých okamžicích mohou mít různé vlastnosti. Při použití čtyřrozměrného modelovacího přístupu jsou běžné předměty chápány jako „časoprostorová čerť“, jsou jen částečně přítomny v každém časovém okamžiku, skládají se tedy z „fází“, v rámci kterých mohou mít různé vlastnosti.

3.2 Základní ontologie

Základní ontologie (v angličtině nazývané např. „foundational“, „top-level“ nebo „upper-level“) obvykle poskytují primární rozlišení veškeré jsoucí skutečnosti do určitých základních kategorií. Poskytují tak základní rámec porozumění mezi komunikujícími agenty (lidskými nebo umělými), zjednodušují vyloučení pojmových nejasností a víceznačností a umožňují explicitní vyjádření ontologických závazků [1].

Základní ontologie mají obecně následující charakteristiky [40]:

- Poskytují referenci pro srovnávání různých ontologických přístupů, a rámec pro analýzu, harmonizaci a integraci ontologií a metadatových standardů.
- Poskytují výchozí bod pro tvorbu nových ontologií. Základní ontologie obsahují předdefinovanou skupinu ontologických entit, které mohou být znovu využity. V ideálním případě také základní ontologie definuje návrhové vzory pro běžně se vyskytující případy modelování.
- Pomáhají uživateli vypořádat se s typickými problémy, na které může při návrhu ontologií narazit.

Podívejme se nyní na některé významné základní ontologie (podrobnější srovnání lze nalézt v [1] nebo [32], a další informace např. v [6,27]):

BFO

Basic Formal Ontology (BFO)⁷ byla vyvinuta týmem B. Smitha na Univerzitě v Lipsku (institut IFOMIS). BFO sestává ze série subontologií, které reprezentují určité perspektivy pohledu na realitu. Nejdůležitější jsou přitom SNAP, což je série ontologií představujících každá katalog entit existujících v daném časovém okamžiku, a SPAN, což je katalog procesů probíhajících napříč časem.

DOLCE

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) je základní ontologie vyvinutá v institutu ISTC-CNR v Římě, původně jako referenční ontologický modul pro knihovnu ontologií v rámci WonderWeb projektu.⁸ DOLCE je založena na metodologii OntoClean [34] a snaží se explicitně vyjádřit své ontologické závazky: vychází z běžného jazyka a snaží se zachytit základní kategorie odpovídající běžnému chápání („common sense“) skutečnosti. Zároveň připouští kolokalizaci entit a používá tak multiplikativní přístup k modelování skutečnosti. DOLCE také obsahuje primitivní relace typu *závislost* nebo *část* a obsahuje rozsáhlou axiomatizaci.

⁷ <http://www.ifomis.uni-saarland.de/bfo/>

⁸ <http://www.loa-cnr.it/DOLCE.html>

Tutoriál

OCHRE

Object-Centered High-Level Reference ontology (OCHRE) je základní ontologie také vyvinutá na ISTC-CNR [41]. OCHRE se snaží zkombinovat deskriptivní přístup založený na běžném chápání skutečnosti („common sense“) s využitím menšího množství základních kategorií a axiomů, používá tedy spíše redukcionistický přístup, z čehož také vyplývá odmítnutí kolokalizace entit. OCHRE je zaměřená na objekty, které mají v jejím ontologickém chápání privilegované postavení, a je založena na extensionalistickém přístupu, tedy předpokládá, že objekty, které mají totožné části, jsou totožné.

OpenCYC

Ontologie OpenCYC⁹ byla vyvinuta v MCC (Microelectronics and Computer Consortium) v rámci projektu CYC, který měl modelovat velkou část běžného chápání skutečnosti. OpenCYC je tou základní částí výsledné ontologie, která je volně dostupná. Celá CYC ontologie pak pokrývá mnoho oborů i konceptů každodenního použití, obsahuje tisíce konceptů. Skupiny konceptů tvoří mikroteorie (kontexty), které se zabývají určitou vymezenou doménou.

SUMO

Suggested Upper Merged Ontology (SUMO)¹⁰ byla vyvinuta skupinou IEEE Standard Upper Ontology (SUO). SUMO nevychází ze specifického teoretického základu. Autoři SUMO vyšli z existujících základních a vyšších ontologií, ty analyzovali, identifikovali základní kategorie, které byli IT komunitou obecně sdílené. Tyto základní ontologie byly syntakticky převedeny na společný formát SUO-KIF, a následně pomocí sémantického sloučení („semantic merge“) byla vytvořena ontologie, která měla určitým způsobem zahrnovat ostatní [30].

KR ontologie

KR ontologie¹¹ byla vyvinuta J. F. Sową a je ovlivněna filosofickými postoji Ch. S. Pierce a A. N. Whiteheada [46]. S ohledem na její ontologické závazky bychom KR ontologii mohli charakterizovat jako revizionistickou a redukcionistickou. Její základní kategorizace je založena na pojmech relace, substance a času. S ohledem na relaci rozlišuje tři základní kategorie entit: *nezávislé*, *relativní*, a *prostředkující* (independent, relative, mediating). S ohledem na substanci rozlišuje dvě základní kategorie: *fyzická* a *abstraktní* (physical, abstract). S ohledem na čas rozlišuje kategorie *endurant* (též continuant) a *perdurant* (též occurent). Na základě těchto základních kategorií vzniká jejich vzájemnou kombinací dvanáct podřazených kategorií, přičemž každá entita spadá do některé z nich.

Následující tabulka přehledně shrnuje hlavní vlastnosti popsanych základních ontologií.

⁹ <http://www.opencyc.org/>

¹⁰ <http://ontology.teknowledge.com/>

¹¹ <http://www.jfsowa.com/ontology/>

	BFO	DOLCE	OCHRE	OpenCYC	SUMO	KR
Deskriptivní	Ne	Ano	Ne	Ano	Ano	Ne
Multiplikativní	Ne	Ano	?	?	?	Ne
Posibilistická	Ne	Ano	Ano	?	?	?
4D	Ano	Ano	Ano	Ano	Ne	Ano

3.3 Doménové ontologie

Jedním z podstatných významových posunů pojmu ontologie bylo jeho rozšíření na modely reality popisující specifické, často i velmi úzce vymezené, věcné oblasti – *domény*. Doménové ontologie vznikají typicky jedním ze dvou způsobů:

- Přímou tvorbou již jako “plnohodnotná” ontologie v určitém formalismu
- Adaptací (v horším případě jen hrubou syntaktickou konverzí) již existujícího, jednodušeji strukturovaného zdroje.

V následujících odstavcích se stručně zmíníme o nejpůvodnějším nebo nejzajímavějších doménových ontologiích z různých oblastí.

Biomedicína

Biomedicínká oblast pravděpodobně zaujímá první místo v míře pokrytí doménovými ontologiemi. Ontologie zde zpravidla vznikají na základě starších terminologických zdrojů, postihujících akumulované zkušenosti generací lékařů.

Relativně nejmenší posun od tradičních zdrojů asi představuje *UMLS*¹² – Unified Medical Language System, vyvíjený Národní lékařskou knihovnou USA. Základní složkou UMLS je *metatezaurus*, který na sebe mapuje pojmy z velkého počtu samostatně vzniklých specializovaných tezaurů včetně jazykových verzí (mezi nimi je např. i česká verze tezauru MeSH). Více „ontologický“ charakter má ovšem jiná část UMLS, *sémantická síť* o 135 sémantických typech („třídách“ entit) a 54 možných vztazích mezi nimi. Každý specializovaný pojem je v UMLS podřazen alespoň jednomu sémantickému typu. Na obr. 1 je část sémantické sítě, s výběrem konceptu „Congenital Abnormality“ (vrozená abnormalita). Je vidět, že vyšší úroveň sémantické sítě jsou jednoduchou obdobou základních ontologií, zatímco nejnižší již mají doménový charakter. V každém případě se však nejedná o ontologii postavenou na striktně logických základech, nýbrž o primárně terminologický zdroj.

Dalším tradičním zdrojem je *Foundational Model of Anatomy*¹³ (FMA) vyvíjený na University of Washington. Obsahuje přibližně 75 tisíc tříd a asi 120 tisíc speciálních termínů. Na obr. 2 je ukázka s podrobnější informací o termínu, „Parietal bone“ (lebeční kost). Můžeme si povšimnout, že FMA již sleduje konkrétnější informace odpovídající situaci entit v reálném světě: odlišuje např. taxonomii tříd a podtříd (levá strana) od taxonomie vztahů celek-část (vpravo nahoře): lebeční kost je částí skeletu hlavy („Skeleton of head“), a její částí je např. lebeční hrbolek („Parietal tuber“); dokonce specifikuje, zda má objekt daného typu nenulový rozměr (což nemají abstraktní entity typu „vývojové stadium“), hmotu (což nemá např. povrch určitého orgánu) a zřetelné ohraničení (což nemají např. látkové entity, jako je krev).

¹² <http://umlsinfo.nlm.nih.gov>

¹³ <http://sig.biostr.washington.edu/projects/fm/>

Ontologické inženýrství

Patrně nejvýznamnějším pokusem konceptualizovat lékařskou problematiku pomocí logických nástrojů je projekt *GALEN*¹⁴. Na rozdíl od předchozích projektů zde nebylo cílem pokrýt vyčerpávajícím způsobem terminologii oboru, nýbrž vytvořit kolekci provázaných definic pojmů, z nichž by se daly (do jisté míry automaticky) vytvářet nové pojmy a vyhledávat nové taxonomické vztahy. Podstatnou součástí modelu jsou proto definice a omezení, vyjádřená v jazyce GRAIL, postaveném na tzv. deskripční logice (viz dále kap. 6.1). Příkladem je definice „odnětí plicního laloku“:

```
(SurgicalDeed which
  isMainlyCharacterisedBy (performance whichG
    isEnactmentOf (Excising which
      playsClinicalRole SurgicalRole) which
        actsSpecificallyOn (Lobe which
          isSpecificSolidDivisionOf Lung))).
```

Z ontologií orientovaných převážně na klinickou medicínu se nakonec zmíníme o kolekci *ON9* [11] vzniklé v rámci projektu ONIONS koordinovaného italskou Laboratoří aplikované ontologie (LOA). V řadě ohledů je obdobou *GALEN*, odlišuje se však důrazem na těsné sepejetí doménové ontologie se základními ontologiemi, a na snahu maximálně využít tradiční zdroje typu UMLS, s tím, že je nutné v nich opravit nekonzistentní místa a tím je převést do podoby „skutečných ontologií“.

V poslední době se v ontologickém inženýrství věnuje velká pozornost aplikacím v oblasti genetiky. Zde je již tradičním zdrojem *Gene Ontology*¹⁵ (GO). Jedná se o poměrně málo strukturovaný, spíše databázově orientovaný zdroj, popisující biologické procesy, buněčné komponenty a molekulární funkce vztahující se k jednotlivým genům.

Podnikání, obchodování, právo

V oblasti podnikání a obchodování lze vysledovat minimálně dva podstatně odlišné typy ontologií. První z nich podrobně strukturují *obchodní komodity* a mají význam zejména pro oblast e-commerce. Druhé se zabývají fungováním *podniku* jako organizace a směřují tudíž k aplikacím uvnitř podnikových informačních systémů. Hlavními zástupci první skupiny jsou produktové katalogy, jako je *UNSPSC*¹⁶, americký *NAICS*¹⁷, německý *E-cl@ss*¹⁸, nebo *RosettaNet*¹⁹. Podobně jako medicínské terminologie jsou velmi rozsáhlé, vznikly původně mimo „znalostní“ prostředí, a jejich „ontologizace“ je předmětem intenzivního výzkumu. Podnikové ontologie reprezentují především *Enterprise Ontology*²⁰ a *TOVE*²¹, obě vzniklé již v 90. letech.

¹⁴ <http://www.opengalen.org>

¹⁵ <http://www.geneontology.org>

¹⁶ <http://www.unspsc.org>

¹⁷ <http://www.census.gov/epcd/www/naics.html>

¹⁸ <http://www.eclass.de>

¹⁹ <http://portal.rosettanel.org/cms/sites/RosettaNet/>

²⁰ <http://www.aiai.ed.ac.uk/project/enterprise/>

²¹ <http://www.eil.utoronto.ca/enterprise-modelling/tove>

Tutoriál

Elektronické obchodování je také hlavní aplikací pro ontologie *webových služeb*. V této oblasti má zřejmě největší vliv komunita okolo ontologie *WSMO*²².

Volně související problematikou jsou *právní* ontologie, které jsou předmětem intenzivního výzkumu [3].

Technika a přírodověda

I v oblasti technického návrhu lze vysledovat rozdělení na rozsáhlé, v praxi používané, ale jednodušeji strukturované zdroje, a menší avšak sofistikované ontologie. Do první skupiny patří především *STEP*²³ jakožto soubor protokolů pro popis produktů včetně procesu jejich výroby. Do druhé skupiny patří například *EngMath* [16] nebo *Phys.Sys* [4], které se věnují vlastnostem a chování obecných fyzických systémů zejména z hlediska prostoročasového a procesního, s cílem podpořit simulační aplikace. Obdobné určení má i *Engineering Ontology* vzniklá v rámci tuzemského projektu *KSMSA*²⁴. V oblasti přírodních věd je pak rozvinuto zejména modelování chemických substancí a procesů [9]. Na rozhraní techniky a přírodovědy stojí kupříkladu modelování problematiky rybnářství [12].

Kulturní dědictví

Vysokou míru zájmu o ontologie projevuje oblast kulturního dědictví. Klíčovou roli zde hraje rozsáhlá ontologie *CIDOC CRM*²⁵, která umožňuje modelovat kategorizaci historických a uměleckých artefaktů i proces jejich vzniku a uchovávání.

Různé

Z dalších oblastí stojí za zmínku aplikace ve *vojenství*, odvíjející se z intenzivní účasti americké vojenské výzkumné agentury DARPA v komunitě znalostního inženýrství. Vojenské ontologie (z nichž některé byly zveřejněny) mapují zejména oblast plánování vojenských operací.

Neméně velký okruh ontologií, zpravidla ovšem vyvíjených veřejně a distribuovaným způsobem, se věnuje tématům souvisejícím se sociálními vztahy a volnočasovými aktivitami. Mezi ně patří zejména nevelká ale velmi využívaná ontologie *FOAF*²⁶, zachycující různorodé informace o lidech a jejich vzájemných vztazích, a perspektivní ontologie *SIOC*²⁷, řešící problematiku sémantického propojení diskusních platform typu elektronických konferencí, blogů apod.

Již řadu let existuje potřeba sémanticky anotovat vedle textových zdrojů také zdroje *multimediální*. Již etablovaným, komerčně šířeným univerzálním standardem v této oblasti je ontologie *MPEG-7*²⁸, která je ovšem spíše rozsáhlým datovým schématem než plnohodnotnou ontologií. V posledních letech jsou stále častější pokusy zkombinovat obsah MPEG-7 s principy ontologického inženýrství: jedním

²² <http://www.wsmo.org>

²³ Viz např. <http://www.wikistep.org>

²⁴ <http://virtual.cvut.cz/ksmsa/>

²⁵ <http://cidoc.ics.forth.gr>

²⁶ <http://xmlns.com/foaf/spec/>

²⁷ <http://sioc-project.org/>

²⁸ <http://en.wikipedia.org/wiki/MPEG-7>

z nejslibnějších výsledků v této oblasti je *COMM*²⁹ – jádrová ontologie multimédií, a na ni navazující ontologie pro různé typy multimediálních dat.

V rámci výzkumných projektů také vznikla také pestrá paleta doménových ontologií, jejichž hlavním účelem jsou, spíše než přímé praktické nasazení, podpora výuky ontologického inženýrství nebo různé teoretické experimenty. Z nich se můžeme zmínit například o ontologii vín³⁰, použité mj. v rámci úvodu do jazyka OWL³¹, nebo ontologii pizz³², na které jsou ilustrována úskalí téhož jazyka pro nezkušeného vývojáře (viz část 6.2); soubor ontologií o problematice pořádání konferencí³³ se zase používá jako testovací pro nástroje mapování ontologií.

4 Jazyky pro reprezentaci ontologií

Z technologického hlediska problematika reprezentace a zpracování ontologií bezprostředně navazuje na tzv. *rámcové* přístupy k reprezentaci znalostí v umělé inteligenci. První rozšířenější jazyk pro reprezentaci ontologií, *Ontolingua* [15], který vznikl na Stanfordské univerzitě počátkem 90. let pod vedením již citovaného T. Grubera, byl původně navržen jako výměnný „mezijazyk“ pro rámcové znalostní systémy. Ve srovnání se současným dominantním zástupcem ontologických jazyků, OWL, měl širší možnosti definování pojmů (využití plného aparátu predikátové logiky), to ovšem vedlo i k obtížím při implementaci odvozovacích nástrojů. Někteří výzkumníci šli následně cestou užšího sepětí s odvozováním v jazyce LISp (který byl v té době syntaktickým základem prakticky všech jazyků pro reprezentaci znalostí) – příkladem je *OCML* [29], který je spíše programovacím jazykem a tvorba ontologií je v něm víceméně nadstavbou. Bohatým jazykem spadajícím do oblasti „tvrdého jádra“ umělé inteligence, ovšem vzhledem ke spojení s komerčními aktivitami používaným omezenou skupinou vývojářů, je *CyCL*, sloužící k tvorbě rozsáhlé všeobecné ontologie CyC³⁴. Alternativou k rámcovým jazykům dále dlouhodobě byly (a v menší míře stále jsou) *konceptuální grafy*, navržené rovněž již citovaným J. Sowou.

Vedle toho se však ve 2. polovině 90. let rozvíjel i směr usilující o užší sepětí ontologií se vznikajícími webovými standardy, jako byl HTML, XML a později RDF. Za historicky první „webový“ ontologický jazyk lze považovat *SHOE*³⁵, ve kterém se ontologie zapisovaly pomocí speciálně pro tento účel navržených (tj. nestandardních) značek HTML. O trochu později vznikaly i jazyky na bázi nativního XML, jako byly *XOL*³⁶ a *OML*³⁷. Už na samém konci 90. let, bezprostředně po publikování (nyní již zastaralé) specifikace jazyka *RDF*³⁸ jako doporučení konsorcia W3C, však diskuse

²⁹ <http://multimedia.semanticweb.org/COMM>

³⁰ <http://www.w3.org/TR/owl-guide/wine.rdf>

³¹ <http://www.w3.org/TR/owl-guide/>

³² <http://www.co-ode.org/ontologies/pizza/>

³³ <http://nb.vse.cz/~svabo/oaei2007/>

³⁴ <http://www.cyc.com/>

³⁵ <http://www.cs.umd.edu/projects/plus/SHOE/>

³⁶ <http://www.ai.sri.com/pkarp/xol/>

³⁷ [http://www.ontologos.org/Simple OML/Simple OML.htm](http://www.ontologos.org/Simple%20OML/Simple%20OML.htm); zajímavostí jazyka je využití konceptuálních grafů (namísto rámcového přístupu) jako formálního základu.

³⁸ Aktuální soubor specifikací jazyka RDF včetně RDF Schema je na <http://www.w3.org/RDF/>.

Tutoriál

odborné komunity o úloze XML versus RDF jako syntaktického rámce pro reprezentaci strojově zpracovatelných dat (faktů) na WWW dokonvergovala k názoru, že RDF je vhodnější zejména z důvodu své modularity a formálně-logické interpretovatelnosti. V zájmu syntaktického sjednocení „webových“ faktů a ontologií definujících jejich význam se posléze jazyk RDF (ovšem sám kanonicky zapisován – serializován – pomocí XML) stal i základní úrovní reprezentace ontologií.

Nejjednodušším ontologickým jazykem nad RDF je *RDF Schema*, který je dnes na úrovni specifikací již propojen s RDF samotným. Umožňuje definovat vztahy podřazenosti mezi třídami zdrojů a také mezi vlastnostmi (predikáty, tj. binárními relacemi definovanými mezi třídami), a dále definiční obor a obor hodnot vlastností.

Na trochu vyšší úrovni složitosti stojí jazyky, jejichž výrazový aparát odráží možnosti vybraných variant *deskripční logiky*. Jedná se především o jazyk *OWL*³⁹, zpracovaný jako specifikace W3C v r. 2004 (jeho předchůdci byly obdobně pojaté jazyky OIL, DAML-ONT a DAML+OIL, publikované postupně od r. 2000), který díky podpoře W3C doznal značného rozšíření, a většina populárních ontologií ho používá přinejmenším jako alternativní způsob zápisu. Oproti RDFS navíc nabízí např. definování tzv. lokálních omezení (existenčních, univerzálních, kardinalitních) nad vlastnostmi vztahenými k určité třídě, možnost postulovat disjunktnost tříd, odlišit nutné a postačující podmínky příslušnosti ke třídě apod. K vyjadřovacímu aparátu jazyků na úrovni OWL se bude vztahovat výklad konkrétních „logických“ problémů v části 6.2, proto jej zde nebudeme podrobně rozebírat. Za zmínku ale stojí, že popularita OWL přesáhla okruh znalostního a webového inženýrství – začala se jím zabývat jedna z klíčových organizací softwarového inženýrství – Object Management Group (OMG). V roce 2003 vznikl na základě výběrového řízení metamodel (rozšíření notace UML) zachycující konstrukce OWL, který by měl umožnit začlenění ontologií jako komponent návrhu rozsáhlejších softwarových aplikací⁴⁰.

Jazykem velmi blízkým OWL, vzniklým de facto v téže odborné komunitě, je *WSML*⁴¹, který je ovšem speciálně navržen pro účel modelování složitých webových služeb. Na úrovni teoretických návrhů se objevují také pokusy o rozšíření ontologických jazyků o reprezentaci neurčitosti. Příkladem je *Fuzzy OWL* [47].

Vedle „středního proudu“ reprezentovaného OWL jako jazykem s vyjadřovací silou (v případě nejběžnější verze jazyka, OWL DL) nižší než má predikátová logika prvního řádu existují i přístupy, které předpokládají vyjadřovací sílu ekvivalentní či vyšší, než má predikátová logika – příkladem je GOL [22].

Jiným způsobem, jak zvýšit vyjadřovací sílu ontologií, je zkombinování ontologického jazyka s na něj navazujícím jazykem *pravidlovým*, který poskytuje komplementární možnosti odvozování. V prostředí sémantického webu je pro tento účel často používán *SWRL*⁴² (Semantic Web Rule Language) navržený v r. 2004.

Na druhé straně je ovšem daleko častější situace, kdy jsou i prostředky deskripční logiky považovány uživateli za příliš komplikované a neintuitivní. V takových případech návrháři sahají buďto po RDFS, nebo po jiných technologiích, které si v určitých komunitách již získaly okruh příznivců. Zajímavá je v tomto směru

³⁹ <http://www.w3.org/2004/OWL/>

⁴⁰ <http://www.omg.org/ontology/>

⁴¹ <http://www.wsmo.org/wsml/>

⁴² <http://www.w3.org/Submission/SWRL/>

Ontologické inženýrství

technologie *Topic Maps*⁴³, sloužící k vytváření a zpracování tzv. *map témat*. Ontologický jazyk *Topic Maps* je v některých aspektech principiálně odlišný od sémantického webu, mimo jiné proto, že mapy témat jsou (v nejrozšířenější variantě notace) zapisovány v nativním XML. Také jejich datový model je ovšem odlišný od RDF, je postaven na množině *témat* (topics) propojených *asociacemi* a majících konkrétní informační zdroje jako svoje *výskyty* (occurrences). Na rozdíl od jazyků ontologií sémantického webu je možné v *Topic Maps* definovat i asociace s aritou vyšší než 2; asociace jsou samy o sobě neorientované (takže nemá smysl rozlišovat dvojici relací vzájemně inverzních, jako v OWL), a k určení pozice zúčastněných témat v asociaci se používají navíc tzv. *role*. Jazyk nemá ambice sloužit pro strojové odvozování, ale jen pro katalogizaci a vyhledávání zdrojů, proto není vázán na formálně-logický aparát. V mnoha ostatních ohledech je plnohodnotnou alternativou k RDF a OWL, komunita jeho uživatelů a vývojářů je ovšem podstatně menší, ale na druhé straně s velkým podílem firem, a tudíž důrazem na jednoduchá, bezprostředně aplikovatelná řešení.

Všechny uvedené jazyky v sobě v zásadě obsahují předpoklad, že třídám ontologie lze přiřadit instance (označované také jako individua, objekty, event. zdroje), tak, že množina instancí podtřídy je vždy podmnožinou množiny instancí nadřazené třídy. Tato množinová sémantika už není chápána jako zavazující u *lexikálních* ontologií (více méně splývajících s tradičními tezaury), kde vztah nadřazeného a podřazeného pojmu může být nejen vztahem třídy a podtřídy, ale také třídy a její instance, případně také celku a části. Pojmy se zde primárně nechápou jako odkazy na množiny objektů reálného světa, nýbrž jako sémantické značky pro slova obsažené v textech. Z hlediska společného reprezentačního jazyka zde můžeme jako jednotný celek zmínit tzv. *wordnety* – jedná se jak o původní anglický *WordNet*⁴⁴ z univerzity v Princetonu, tak i o velký počet národních *wordnetů* sdružených v mezinárodní asociaci GWA⁴⁵: mezi velmi dobře zpracované jazyky patří, díky dlouhodobé práci týmu z Masarykovy univerzity, i čeština.

Za účelem dosažení alespoň částečné kompatibility tezaurových zdrojů s ontologiemi zapsanými v jazycích typu RDFS resp. OWL pak vznikl *SKOS*⁴⁶ (Simple Knowledge Organisation System); jazyk je sám definován jako ontologie pomocí RDFS, a tudíž je i možné ho s jinými ontologiemi kombinovat. Jde o odlehčený ontologický jazyk přizpůsobený zdrojům charakteru tezauru, umožňuje definovat např. vztah mezi pojmem a jeho slovní definicí, vztah obecnějšího a speciálnějšího pojmu (pomocí relací typu `skos:broader`, tj. nikoliv jako třídu a podtřídu).

Za extrémně jednoduché „ontologie“ jsou někdy považovány i ad hoc hierarchie, např. tématické taxonomie používané ve znalostním managementu v podnikovém prostředí nebo struktury záhlaví ve webových katalogích typu Yahoo! či Seznam. Zde ovšem není systematicky zohledněna sémantika ani množinová ani lexikální, a jediným styčným bodem se „skutečnými“ ontologiemi je přítomnost hierarchické struktury pojmů, což je pro adekvátnost používání označení „ontologie“ málo.

⁴³ <http://www.topicmaps.org/>; viz též český tutoriál J. Koska na <http://www.kosek.cz/xml/tmtut/>

⁴⁴ <http://wordnet.princeton.edu/>

⁴⁵ <http://www.globalwordnet.org/>

⁴⁶ <http://www.w3.org/2004/02/skos/>

5 Prostředky pro podporu ontologické konceptualizace

5.1 Stručný přehled

Ačkoliv se na první pohled může zdát, že pečlivé zpracování specifikací ontologického jazyka a jeho podpora odvozovacími nástroji je postačující podmínkou pro vznik prakticky využitelných ontologií, ve skutečnosti tomu tak není. Ontologický artefakt, který je z formálního hlediska zcela v pořádku, může být zcela neadekvátní z hlediska obsahu. V [52] jsou charakterizovány tři klíčové aspekty adekvátnosti ontologického obsahu:

- *Přesnost*: obsah ontologie by měl co nejméně odpovídat situacím v reálném světě. Mnoho ontologických modelů je oproti realitě natolik zkreslených či zjednodušených (ať už vinou nedostatečně pečlivé analýzy reality, nebo kvůli omezením reprezentačního jazyka), že formálně správná odvození nad nimi vedou k nesmyslným závěrům.
- *Srozumitelnost*: ontologie by měla být co nejlépe srozumitelná lidem, kteří se na jejím vývoji nepodíleli. Zde má velký význam vhodné pojmenování prvků ontologie (např. vyvarování se žargonu), ale také modularizace na části, které je člověk schopen rozumově obsáhnout.
- *Způsobnost k odvozování*: nad ontologií (pokud má mít vyšší ambice než tradiční tezaury a klasifikační schémata) by mělo být možné netriviálně strojově odvozovat. Pro to je třeba využít možností formálního aparátu jazyka.

Mezi těmito aspekty lze vyzorovat různé pozitivní i negativní vazby. Snaha o bohaté odvozovací možnosti nad ontologií někdy vede k přidávání vztahů, které platí pouze "modelově" a nikoliv paušálně⁴⁷, čímž se snižuje přesnost. Podobně i snaha o maximální srozumitelnost pro nezasvěcené uživatele může vést ke snižování přesnosti (např. pokud se přesné avšak málo známé pojmy nahradí běžnými ale nepřesnými). Na druhé straně vysoká míra formalizace (pro potřebu odvozování) může napomoci odhalit již ve fázi úvodního návrhu některé odchylky např. taxonomické kostry oproti realitě.

V následujícím textu se podrobněji zmíníme o několika typech prostředků, které se v posledních letech používají jak podpora vzniku ontologického obsahu relativně nezávisle na konkrétním jazyce formalizace: o strukturovaných metodikách, o využití základních ontologií (resp. s nimi úzce spojených obsahových vzorů) jako východiska pro tvorbu ontologií doménových, a o automatické analýze podkladových materiálů (někdy označované jako „učení ontologií z textů“).

5.2 Strukturované metodiky pro vývoj ontologií

Podobně jako v softwarovém inženýrství a později i ve znalostním inženýrství chápaném jako tvorba znalostních softwarových aplikací [42] se potřeba obecných doporučených postupů práce objevila i v ontologickém inženýrství jakožto specifické

⁴⁷ To lze částečně řešit pomocí rozšíření jazyků o formalizaci neurčitosti.

Ontologické inženýrství

součástí znalostního inženýrství zaměřené na dílčí „datově-sofwarové“⁴⁸ artefakty – ontologie. [13] popisuje a porovnává sedm obecných metodik pro vývoj ontologií, převážně vzniklých zobecněním zkušeností z jednoho či více konkrétních projektů. V následujícím textu se pokusíme zobecnit posloupnost hlavních kroků, která se objevuje ve většině metodik.

Ujasnění účelu a rozsahu ontologie

Na začátku je potřeba vymezit, jaké úlohy budou s pomocí ontologie řešeny, a jaká část dané věcné oblasti má být modelována. To je způsob, jak překonat tzv. *problém rozsahu* („hugeness problem“), tj. skutečnost, že k libovolné věcné doméně se zpravidla vztahuje (třebas okrajově) nezvládnutelné množství pojmů a vztahů.

Vedle obecných *scénářů* či *případů použití*, pojatých obdobně jako v softwarovém inženýrství „konvenčních“ aplikací, stojí za zmínku tzv. *kompetenční otázky* [17]. Jedná se o příklady konkrétních otázek, které by hypotetický znalostní systém s pomocí vyvíjené ontologie měl být schopen zodpovědět.

Je ovšem potřeba si uvědomit, že ontologie by neměla být jednoúčelová. Měla by sice každopádně být navržena tak, aby vyhověla potřebám úlohám řešeným v rámci daného projektu, na druhou stranu by však měla předjímat i „rozumně velké“ spektrum dalších úloh z téže domény, tj. zůstat otevřená pro „znovupoužití“ (reuse).

Specifikace terminologické části ontologie

Přestože znalostní ontologie svým pojetím sémantiky přesahují čistě terminologický pohled, vhodným výchozím bodem pro jejich vlastní tvorbu je seznam relevantních *termínů*, prozatím ještě bez rozčlenění na třídy, relace, instance atd. Prostý seznam termínů je ale možné doplnit jejich vysvětlením, event. definicemi, formou volného textu. Tím se ze seznamu stane *glosář*, který může později sloužit i jako dokumentace dané ontologie pro člověka.

Přínejmenším v této fázi bývá efektivní použít prostředky pro automatickou analýzu podkladových materiálů, kterým se věnuje podkapitola 5.4.

Odlišení ontologických typů

Principiálním rozdílem mezi slovníky/glosáři a byť jen jednoduchými ontologiemi je explicitní rozlišování ontologických typů, kterými jsou zejména třídy, relace (plus atributy) a instance tříd. Zatímco v některých případech volba víceméně jednoznačně vyplývá z povahy věci, jindy je možné tentýž objekt či vztah modelovat více způsoby. Již v této fázi je proto třeba zohlednit cílovou aplikaci; pomůckou mohou být logické vzory, viz část 6.2.

Specifikace taxonomie

Taxonomická struktura je ve většině ontologických jazyků chápána jako kostra, na kterou se teprve navazují další vztahy. Taxonomii je možné budovat *odshora* (top-down), tj. od nejobecnějších pojmů dané domény, pokud možno opřených o základní

⁴⁸ Tato dualita pohledu na ontologie je poměrně významná. Tvorba ontologie se na jedné straně podobá návrhu datového modelu, na druhé straně kódování jejich axiomů ve formálním jazyce nemá daleko k programování v jazycích typu Prolog nebo LISp.

Tutoriál

ontologie, viz níže, nebo také *odspoda* (bottom-up), tj. od nejkonkrétnějších pojmů (např. individuových jmen), za nevhodnější se však ve většině případů považuje postup *od středu* (middle-out). Při tomto přístupu se jako výchozí bod vezmou „středně obecné“ doménové pojmy, protože ty mají největší šanci být použitelné pro velkou řadu aplikací; k nim se pak přidávají pojmy obecnější nebo naopak konkrétnější, ovšem pouze tehdy, když jsou skutečně potřeba a nelze je substituovat „středně obecnými“ pojmy.

Ve fázi tvorby taxonomie je také nutné vyřešit otázku importu již existujících doménových ontologií, a také mapování na *základní ontologie*, kterému se věnuje podkapitola 5.3.

Vytvoření struktury netaxonomických relací, atributů a instancí

Na základní taxonomickou strukturu se postupně navazují netaxonomické relace mezi třídami. Důležité je zde zvolit vhodnou hierarchickou úroveň, ke které se relace budou vztahovat. Totéž platí i pro atributy tříd⁴⁹. Instance se ve většině případů do ontologií zařazují pouze tehdy, když mají v dané oblasti významné, jedinečné postavení. Některé návody a metodiky postulují, že instance by měla být do ontologie zařazena pouze tehdy, když je nutná k definování určité třídy⁵⁰.

Nasazení a údržba ontologie

Vedle ústředního procesu vlastní tvorby ontologie se do jejího životního cyklu popsaného metodikami často zahrnuje, tak jako u jiného softwarového artefaktu, fáze jejího nasazení a údržby, která má samozřejmě odlišný časový horizont než předchozí fáze. Některé metodiky zde např. odkazují na metody testování a verzování ontologií.

Poznamenejme, že některé „důkladnější“ metodiky, jako je METHONTOLOGY [13], předpokládají, že ontologie zpočátku vzniká v podobě neomezené (nebo jen málo omezené) cílovým logickým formalismem. Jednotlivé konstrukty i vztahy mezi nimi jsou zaznamenávány do tabulek, a logické formule formou pseudojazyka (např. s vyjadřovací silou predikátového kalkulu), nad kterým se ještě nepředpokládá odvozování. První verze ontologie pak tedy může existovat na úrovni textového dokumentu; některé softwarové nástroje, jako (v současnosti ovšem již neudržovaný) WebODE⁵¹, specificky podporují takovou „reprezentačně nezávislou“ tvorbu ontologií, a umožňují následnou konverzi do různých formátů. Zcela automatická konverze by však vyžadovala sofistikované zohlednění logických vzorů pro cílový jazyk (viz část 6.2), jaké v současnosti ještě v žádném systému zřejmě implementováno není.

⁴⁹ V některých jazycích je mezi relacemi a atributy úzký vztah. Příkladem je OWL, kde se označují za objektové resp. datotypové vlastnosti, a mají některé společné charakteristiky. Datotypové vlastnosti v OWL mohou, stejně jako objektové, nabývat více hodnot, protože se jedná v obou případech o binární relace, ne nutně funkce.

⁵⁰ Příkladem takové definice je (ve slovním vyjádření) např.: „X je instancí třídy *anglicky-hovořící-člověk* právě tehdy, když je pro něj vlastnost *mluví-jazykem* nabývá hodnoty *angličtina*“, přičemž individuum *angličtina* je instancí třídy *jazyk*.

⁵¹ <http://webode.dia.fi.upm.es/WebODEWeb/index.html>

Metodiky, které podporují spíše přímé kódování ontologie, pak naopak samozřejmě postulují průběžné nebo alespoň dodatečné zdokumentování ontologie a procesu jejího vzniku.

5.3 Aplikace základních ontologií a obsahových vzorů

Návrh ontologií je náročnou činností, která vyžaduje jak rozsáhlé znalosti v oblasti modelování, tak i porozumění dané předmětné doméně, která je modelem zachycována. Pro usnadnění procesu modelování je možné využít, vedle výše popsaných základních ontologií, i další postupy. Významná je zejména specializovaná metodologie *OntoClean*, jejímž cílem je omezit výskyt nejběžnějších modelovacích chyb, a zajistit konzistenci výsledné ontologie. Dalším nástrojem jsou *návrhové vzory*, které umožňují podobně oblastí modelovat podobným způsobem, na základě důkladně propracovaných vzorových přístupů a šablon.

OntoClean je metodologie pro ontologickou analýzu vyvinutá N. Guarinem a C. Weltem [20]. Vychází z obecně teoretických výzkumů v oblasti filosofické ontologie a aplikuje je s cílem zajistit, aby ontologie, která je výsledkem modelování, byla konzistentní a neobsahovala zásadní konceptuální chyby. *OntoClean* tak pak vychází z určitých abstraktních ontologických pojmů, na základě nichž definuje specifické *metavlastnosti*, pomocí nichž pak popisuje jednotlivé součásti navrhované ontologie. Přirazené *metavlastnosti* pak jednak omezují další vývoj ontologie pouze na s ohledem na *OntoClean* konzistentní rozšíření, a zároveň umožňují vykázat existující nekonzistence v dané ontologii. Podívejme se tedy na několik základních ontologických pojmů, se kterými *OntoClean* pracuje.

Prvním takovým pojmem je *esencialita*, a týká se vztahu mezi entitou a její vlastností. Vlastnost je pro entitu esenciální, pokud ji daná entita musí mít. Speciálním typem esenciality je *rigidita*. Vlastnost je rigidní, pokud je esenciální ve všech svých případech. Například vlastnost „být osobou“ je obvykle považována za rigidní, neboť pokud ji kterákoliv entita, která ji má, ztratí, přestane být tím, čím je (například smrtí se ze člověka stane mrtvola). Naopak vlastnost „být tvrdý“ není rigidní, neboť různé entity ji mohou nabývat a ztrácet, aniž by přestaly být tím, čím jsou (například mokrá houba je měkká, po usušení je tvrdá, přesto je to stále houba). Vlastnosti, které v některých případech jsou esenciální a v jiných ne, jsou nazývány *semirigidní* a vlastnosti, které nejsou nikdy esenciální, jsou nazývány *antirigidní*. Při aplikaci metodologie *OntoClean*, je všem vlastnostem v ontologii přiřazena jedna z těchto *metavlastností* – rigidita, semirigidita nebo antirigidita. Výhodným výsledkem je následně i jednoduchá identifikace tzv. *páteří taxonomie* dané ontologie, která sestává ze všech rigidních vlastností.

Dalšími důležitými pojmy, kterými se *OntoClean* zabývá, jsou pojmy identity a jednoty. Problém *identity* je otázka, v jakém případě považovat dvě entity za identické. Klasickým případem je identita Jitřenky a Večernice, jiným příkladem je aplikace sledující kamerami určitý prostor a nutnost rozhodnout se, kdy člověk zabíraný kamerou A je identický s člověkem, který byl před chvílí snímán kamerou B v jiné lokaci. Ve všech takových případech je podstatné explicitní vyjádření tzv. kritéria identity, tj. podmínek, které dvě entity musí splňovat, aby byly prohlášeny za identické. Obecně předpokládáme, že každá entita v ontologii by měla mít určité kritérium identity a rigidní vlastnosti, které toto kritérium identity popisují. Téma

Tutoriál

jednoty se týká otázky, v jakém vztahu jsou části určité entity k celku. Problémovým konceptem je třeba „voda“, jehož instance jsou určitá množství vody, ale obvykle nejsou považována za nějaké jednoty nebo vymezené celky. Pokud naopak chápeme například Černé moře jako určitý jeden vymezený celek, můžeme jej chápat jako instanci konceptu „moře“, ale dále už nemůžeme „moře“ jednoduše považovat za podtřídu třídy „voda“. Tento problém podle autorů OntoCleanu vychází z mnohoznačnosti jazyka, protože moře ve skutečnosti nejsou druhy vod, ale jsou složeny z vody. Proto metodologie OntoClean rozlišuje prostřednictvím metavlastností ty koncepty, jejichž instancemi jsou nějaké jednotné entity, celky, a ty, u nichž tomu tak není.

OntoClean také věnuje pozornost vztahu subsumpce mezi koncepty (resp. podtřídy mezi třídami), který je základním vztahem pro většinu ontologií a často se v jeho použití chybí. Základní charakteristikou tohoto vztahu je, že všechny instance podřazeného konceptu jsou nutně i instancemi konceptu nadřazeného. Typický problém je zaměňování vztahu subsumpce a instanciace. Autoři OntoCleanu uvádějí následující příklad: „Člověk“ je podtřídou třídy „Savec“ a „Sokrates“ je instancí třídy „Člověk“. Jak je to ale se vztahem třídy „Biologický druh“ k třídě „Člověk“? Často se stává, že nezkušený návrhář ontologie určí třídu „Člověk“ jako podtřídu třídy „Biologický druh“. Pokud ovšem určíme kritérium identity pro entity spadající pod třídu „Člověk“ a kritérium identity pro entity spadající pod třídu „Biologický druh“ zjistíme, že jsou zásadně odlišná. Pokud by „Člověk“ byl podtřídou „Biologického druhu“ pak by jistě zdědil jeho kritéria identity. Ve výsledku docházíme k závěru, že „Člověk“ je instancí třídy „Biologický druh“, a to je třída druhého řádu.

Dalšími problémovými oblastmi návrhu ontologií, kterým se metodologie OntoClean věnuje, jsou záměny vztahu subsumpce a vztahu část-celek, problematika polysémie a další.

Metodologie OntoClean je v současnosti používána v praxi, a existují softwarové nástroje pro podporu jejího nasazení. Například pro jeden z nejpoužívanějších ontologických editorů – Protégé – existuje modul pro doplňování metavlastností definovaných metodologií OntoClean.

Obsahové vzory

Obsahové návrhové vzory – Conceptual (or Content) Ontology Design Pattern (CODeP) – jsou nástrojem, šablonou pro zjednodušení a standardizaci řešení modelovacích problémů při návrhu ontologií. A. Gangemi [10] definuje některé charakteristické vlastnosti obsahových návrhových vzorů:

- Návrhový vzor je obvykle vytvořený na základě určité referenční základní ontologie, ale je aplikovatelný i v jiných ontologiích postavených na odlišných základních ontologiích.
- Návrhové vzory jsou obvykle dobře vizualizovatelné, a jsou přiměřeného rozsahu, aby byly snadno zapamatovatelné a lehce využitelné. Návrhové vzory mohou využívat jeden druhý nebo mohou vytvářet hierarchii podle úrovně zobecnění.
- Konkretizované návrhové vzory pro danou doménu mohou být vytvořeny doménovými experty s použitím obecnějších návrhových vzorů a obecných doménových schémat.

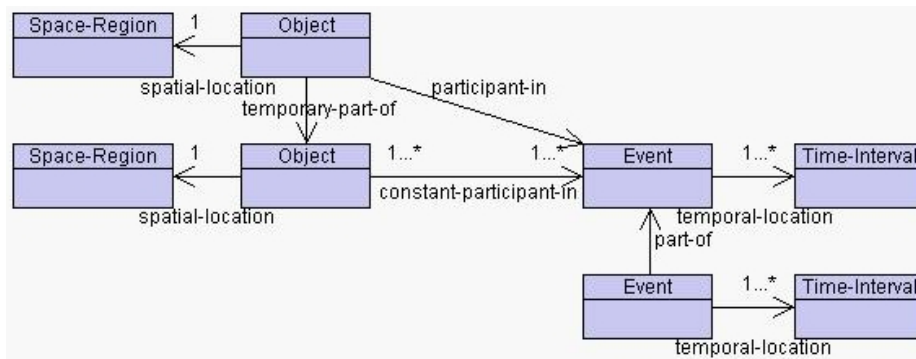
Ontologické inženýrství

- Návrhové vzory mohou také zachycovat doporučené postupy („best practices“) pro určité oblasti modelování. Návrhové vzory mohou připomínat databázové schéma, ovšem na rozdíl od nich jsou definovány s ohledem na referenční ontologii a mají obvykle obecnější charakter.

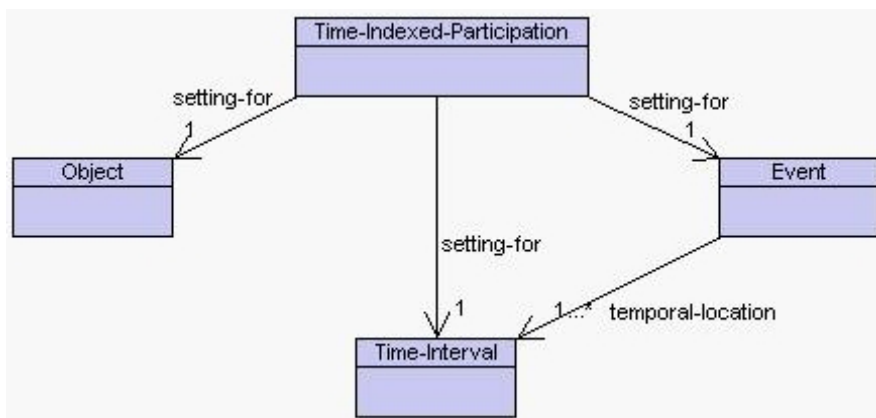
Jako nejběžnější příklady obsahových návrhových vzorů uvádí [10] modelování přítomnosti v časoprostorové lokaci a reifikaci časově indexované relace.

Jádro prvního vzoru, který je součástí knihovny základní ontologie DOLCE [26], tvoří *relace účasti* (participace), která je mezi objektem a událostí. Časová indexace je zajištěna lokalizací dané události v určitém časovém intervalu. Vzor poskytuje možnost hierarchizace objektů i událostí, umožňující podporu jednoduchého odvozování. Pokud je objekt účasten události, jsou účastny události i jeho části. Pokud je objekt trvale účasten nějaké nadřazené události, je účasten i událostí, které jsou časovými částmi této nadřazené události. Viz Obr. 3.

Obsahový vzor *reifikace časově indexovaných relací* (reifikace – zvětčení, z lat. „res“ – věc, „facere“ – učinit) představuje určitou alternativu pro první příklad. Abstraktní koncept *Time-Indexed-Participation* zvětčuje, tj. vytváří entity reprezentující vztah mezi objektem, událostí a časovým intervalem. Viz Obr. 4.



Obr. 3: Přítomnost v časoprostorové lokaci.



Obr. 4: Reifikace časově indexovaných relací.

Můžeme tedy pro danou konkrétní situaci často vybrat z více aplikovatelných návrhových vzorů ten nejvhodnější. Pro vytváření knihoven návrhových vzorů je doporučena popisová šablona, k zachycení podstatných charakteristik každého návrhového vzoru [10].

5.4 Automatická analýza podkladových materiálů

Tvorba ontologie byla tradičně považována za sofistikovanou a tudíž čistě lidskou aktivitu. Přibližně od roku 2000 se však začalo ve větší míře uvažovat o zapojení automatických metod i do této oblasti. Důvodem je zejména vysoká časová náročnost a možnost ovlivnění subjektivním pohledem na doménu při vymezení okruhu relevantních termínů jako úvodní fázi vlastní tvorby ontologie. Pokud jsou pro danou oblast k dispozici rozsáhlejší textové zdroje, je možné v nich detekovat *relevantní termíny* metodami známými z dokumentografických informačních systémů (běžně se pro tento účel používá např. míra TDIDF). Druhou možností, která funguje spíše v úzce vymezených oborech, je využití syntaktických vzorů nad (povrchovou či hloubkovou) větnou strukturou; např. pokud se v lékařském textu vyskytne určité podstatné jméno bezprostředně před slovesem „postihuje“, můžeme předpokládat, že se jedná o slovo, které bude kandidátem na zařazení do ontologie.

Ambicióznější metody *učení ontologií* [5,25] jdou pak ještě dál: jejich cílem je nabídnout uživateli na základě nejen seznam termínů, ale také uspořádat je do hierarchické struktury (např. ve výše uvedeném příkladě by se nové slovo mohlo označit za kandidáta na podtřidu či instanci třídy Nemoc), nalézt potenciální relace mezi pojmy, případně i navrhnout složitější logické axiomy. Míra přesnosti výsledků takových automatických metod je zpravidla nepřilíš vysoká, pro tvůrce ontologie však přesto mohou představovat užitečný vstup.

6 Ontologie a logické odvozování

Web Ontology Language (OWL) má tři odlišné standardizované podoby – Full, DL a Lite. OWL Full označuje „kompletní“ jazyk OWL, zatímco verze DL a Lite jsou určitým způsobem redukovány a mají tudíž i nižší expresivní schopnost. OWL Lite je přitom redukována oproti OWL DL a je určena pro aplikace s požadavkem na jednoduchou implementaci a obsahuje pouze základní sadu konstruktů jazyka pro nejjednodušší použití. OWL DL je jazyk korespondující s *deskripční logikou*, která je teoretickým podkladem pro algoritmizované odvozování v oblasti ontologií.

6.1 Odvozování v deskripční logice

Deskripční logika

V následujícím oddíle podáme základní neformální přehled deskripční logiky – zájemce o přesné formální vymezení odkazujeme na standardní příručku [2]. Deskripční logika je logika *pojmu* neboli *konceptů*. Primárně tedy zachycuje vzájemné vztahy mezi pojmy. Pojem můžeme zjednodušeně chápat jako obecný výraz, pod který spadá či může spadat více individuí. Příkladem jsou výrazy jako

Ontologické inženýrství

„židle“, „nábytek“, „otec“, „planeta“, „muž“, „žena“ atd. Pojmy odlišujeme od *vlastních jmen*, která označují jedno konkrétní individuum, např. „Václav Klaus“. Koncepty pak ze sémantického hlediska můžeme chápat jako množiny individuí, spadajících pod příslušný pojem. Můžeme tedy také vyslovit tvrzení o tom, že určité individuum spadá pod určitý koncept: *Muž(Petr)*, tedy že Petr je muž.

Ze základních konceptů (např. *Muž* a *Rybář*) je možno konstruovat složené koncepty. Například koncept $Muž \sqcap Rybář$ zahrnuje ta individua, která jsou zároveň mužem a rybářem (sémanticky se jedná o průnik příslušných množin individuí). Analogicky koncept $Muž \sqcup Rybář$ zahrnuje ta individua, která jsou buď mužem anebo rybářem (sémanticky se jedná o sjednocení příslušných množin). Koncept $\neg Muž$ zahrnuje ta individua, která nespádají pod koncept *Muž*.

Druhou základní součástí deskripční logiky jsou *role*. Role můžeme naopak sémanticky chápat jako množiny uspořádaných dvojic, tedy jako binární relace. Příkladem rolí pak mohou být *obíháKolem* (např. Země obíhá kolem Slunce), *jeOtcem* (např. Karel je otcem Petra). Formálně pak můžeme příslušné fakty zapsat například: $obíháKolem(Země, Slunce)$ a $jeOtcem(Karel, Petr)$. S použitím role *obíháKolem* můžeme zkonstruovat další koncepty, například $\exists obíháKolem.Hvězda$, což je koncept planety (pro každou planetu existuje hvězda, kolem které obíhá), nebo $\forall jeOtcem.Muž$ což je koncept zahrnující všechna individua, jejichž všechny děti⁵² jsou mužského pohlaví. Všimněme si, že symboly \forall a \exists mají jiný význam než ve standardní predikátové logice, nicméně formulí deskripční logiky lze na první pohled odlišit od formule predikátové logiky podle syntaxe. Nehrozí tedy záměna.

Formálně se konstruktor \forall z hlediska syntaxe deskripční logiky používá ke konstrukci výrazů typu $\forall R.C$, kde R je role a C je koncept. Z hlediska sémantiky (vyjádřeno v jazyce predikátové logiky) popisuje tento koncept množinu $\{x; \forall y (R(x,y) \rightarrow C(y))\}$. Tedy v případě výrazu $\forall jeOtcem.Muž$ se jedná o takovou množinu prvků x , kde všechna y , pro která platí $jeOtcem(x,y)$, spádají pod koncept $Muž(x)$. Což znamená, jak už bylo řečeno, že se jedná o množinu všech otců dětí jen mužského pohlaví. Analogicky se konstruktor \exists z hlediska syntaxe používá ke konstrukci výrazů typu $\exists R.C$, kde R je role a C je koncept. Z hlediska sémantiky popisuje tento koncept množinu $\{x; \exists y (R(x,y) \wedge C(y))\}$. Tedy v případě výrazu $\exists obíháKolem.Hvězda$ se jedná o takovou množinu prvků x , pro které existuje y , kde platí zároveň $obíháKolem(x,y)$ a $Hvězda(y)$. Tedy jedná se o množinu objektů, které obíhají kolem nějaké hvězdy, tzn. definujeme koncept planety (předpokládáme, že objekt může v daný okamžik obíhat jen kolem jednoho objektu, a za univerzum diskursu považujeme jen velká vesmírná tělesa). Univerzální koncept \top zahrnuje všechna individua z univerza diskursu, prázdný koncept \perp nezahrnuje žádné individuum.

Deskripční logiku můžeme vlastně chápat jako určitý fragment predikátové logiky prvního řádu, používající odlišnou syntaxi – predikátová logika prvního řádu je expresivnější než deskripční logika. Deskripční logika je na rozdíl od predikátové logiky *rozhodnutelná*. Deskripční logika přitom není jen jedna, rozeznáváme celou rodinu deskripčních logik s různou expresivitou, zachovávající si však rozhodnutelnost. Základní deskripční logika *AL* připouští pouze následující

⁵² Přesněji: všechny děti, kterým jsou otcem, protože matky koncept nezahrnuje.

Tutoriál

konstrukty: $\neg A, C \sqcap D, \forall R.C, \exists R.C, \top$, kde R je atomická role a A je atomický koncept a C, D jsou koncepty.

Znalostní báze deskripční logiky je tvořena TBoxem a ABoxem (někdy se mluví ještě o RBoxu obsahujícím axiomy rolí). TBox obsahuje *terminologické axiomy* formátu $A \equiv C$ nebo $A \sqsubseteq C$ (např. *Člověk* \equiv *Živočich* \sqcap *Rozumný*), dále axiomy týkající se rolí formátu $R \equiv S$ nebo $R \sqsubseteq S$ (přípustný až logice *SH*). ABox obsahuje extenzionální axiomy týkající se spadání individuí pod koncepty a účasti v rolích, například $C \sqcap D(a)$ nebo $R(a,b)$, kde a,b jsou individua.

Další rozšíření základní deskripční logiky se vyznačují písmenem – přípustné konstrukce shrnuje následující tabulka (konkrétní deskripční logika se pak označuje kombinací písmen, např. *ALCN*):

Označení	Přípustná rozšíření
U	Sjednocení $C \sqcup D$
E	Plný existenční kvantifikátor $\exists R.C$
N	Numerické restriky (kardinalitní omezení) $\leq nR$ a $\geq nR$
C	Plná negace $\neg C$

Symbol S se označuje deskripční logika ALC_{r+} , tzn. deskripční logika *ALC* rozšířená o axiomy tranzitivity rolí. Symbolická označení pokročilých deskripčních logik shrnuje další tabulka:

Označení	Přípustné rozšíření
S	$S = ALC +$ axiomy tranzitivity rolí
SH	$SH = S +$ axiomy hierarchie (inkluze) rolí
SHf	$SHf = SH +$ funkcionální axiomy rolí
SHO	$SHO = SH +$ nominálové axiomy
$SHOI$	$SHOI = SHO +$ inverzní role
$SHOIN$	$SHOIN = SHOI +$ numerické restriky
$SHOIQ$	$SHOIQ = SHOI +$ kvalifikované numerické restriky

Transitivní role nesmí být použity v numerických restrikcích, jinak je výsledná deskripční logika nerozhodnutelná. Množiny transitivních rolí a funkcionálních rolí by mely být disjunktní.

Nyní můžeme konstatovat, že OWL DL odpovídá deskripční logice *SHOIQ*. V současné době je připravována verze OWL 1.1, která má přinést řadu rozšíření, a z teoretického hlediska bude odpovídat expresivnější deskripční logice [14].

Základní odvozovací úlohy v deskripční logice jsou následující:

- *Kontrola konzistence* znalostní báze zjišťuje, zda definice každého konceptu připouští náležení alespoň jednoho individua.
- *Kontrola individuí* zjišťuje, zda dané individuum spadá pod daný koncept.
- *Realizace* nalezne nespécifičtější koncept, pod který individuum spadá.
- *Vyhledání* nalezne všechna individua spadající pod zadaný koncept.
- *Kontrola subsumpce* zjišťuje, zdali je jeden zadaný koncept podkonceptem druhého.

Většina odvozovacích systémů je v současnosti založena na *tablových* (tableau) algoritmech. Pro OWL ontologie je k dispozici řada funkčních odvozovacích softwarových nástrojů, např. Pellet [43], Racer [21] nebo FaCT++ [53].

6.2 Strukturální a logické problémy při tvorbě ontologií

Jak už jsme naznačili v části 5, tvorba obsahu ontologie skrývá pro běžného informatika některá úskalí, o kterých se zmíníme v následující části. Nejprve se dotkneme jednodušších jmenných a strukturálních konvencí, a poté se budeme věnovat hlubším problémům logické interpretace ontologií specificky pro jazyk OWL, a specificky vytvořeným tzv. logickým návrhovým vzorům.

Struktura a pojmenování tříd, relací a instancí

Přestože s požadavkem, že všechny instance dané třídy by měly být současně instancemi každé z jejích nadtříd, každý začínající ontologický inženýr ochotně deklaruje souhlas, je skutečností, že v málokteré ze začátečnických (a leckdy dokonce i profesionálních) ontologií je požadavek bezevbytku splněn⁵³. Často je za vztah třídy a podtřídy vydáváno to, co je ve skutečnosti vztahem třídy a instance (třeba Firma – IBM) nebo vztahem celku a části (třeba Počítač – Procesor). Někdy je také vztah samotný zamýšlen dobře, ale není vhodně realizován na úrovni pojmenování tříd, protože předpokládá interpretaci názvu třídy v kontextu nadřazených tříd (třeba Počítač – Stolní).

Dalším zdrojem nejasností je pojmenování relací, které neevokuje směr relace. Pokud relaci vyjadřující vztah vyrábějící firmy a vyráběného produktu nazveme třeba „výrobek“, nemusí být nezasvěcenému člověku jasné, zda tvrzení „IBM výrobek ThinkPad“ znamená „IBM vyrábí výrobek ThinkPad“ nebo „IBM je výrobek (firmy) ThinkPad“. Vhodnější je tedy rozlišovat vzájemně inverzní relace třeba takto: „je_výrobkem“ / „má_výrobek“, tj. hlavním slovem názvu relace by mělo být sloveso, eventuelně předložka.

Pro lepší čitelnost složitějších axiomů je navíc dobré od sebe třídy, relace a případně i instance odlišit typografickou konvencí. Často se alespoň názvy tříd píšou s velkým počátečním písmenem.

Logické „paradoxy“

Pro informatiky zvyklé na „databázové vidění světa“ může tvorba ontologie v OWL DL skrývat další úskalí, která propagátoři ontologií z iniciativy CO-ODE⁵⁴ nápaditě zviditelnili pomocí demonstrační ontologie „pizzas“ a vysvětlujících materiálů k ní.

Prvním problémem je chápání existenčního a univerzálního kvantifikátoru, \forall a \exists (v syntaxi OWL „allValuesFrom“ a „someValuesFrom“). Poměrně nasnadě je asi nesprávnost definice vegetariánské pizzy jako pizzy s vegetariánskou náplní⁵⁵ (kde třída *Vegetariánská_náplň* zahrnuje podtřídy jako *Rajčatová_náplň*, *Houbová_náplň*

⁵³ O problému jsme se zmínili již v části 5.3 v souvislosti s metodou OntoClean. Je však natolik zásadní, že není na škodu na něj upozornit na více místech.

⁵⁴ <http://www.co-ode.org/>

⁵⁵ Nepříliš výstižný avšak v praxi používaný překlad anglického „topping“; kvůli němu jsou některé uvedené příklady pro českého čtenáře drobně modifikovány.

Tutoriál

atd.), tj. $Vegetariánská_pizza \equiv Pizza \sqcap \exists má_náplň.Vegetariánská_náplň$. Pizza totiž (pokud není relace $má_náplň$ deklarována jako funkce) může mít náplní víc, definice ovšem pouze požaduje, aby byla vegetariánská aspoň jedna z nich. Někoho ale možná zaskočí, že problém nevyřeší prostá výměna existenčního kvantifikátoru za univerzální, čili: $Vegetariánská_pizza \equiv Pizza \sqcap \forall má_náplň.Vegetariánská_náplň$. Pojem vegetariánské pizzy totiž pak zahrnuje kromě opravdových vegetariánských pizz také pizzy, které nemají žádnou náplň (a tudíž lze říci, že všechny náplně, které mají, jsou vegetariánské...). K dosažení žádoucího stavu je tedy nutno na pravé straně definice uvést konjunkci existenčního a univerzálního axiomu: $Vegetariánská_pizza \equiv Pizza \sqcap \exists má_náplň.Vegetariánská_náplň \sqcap \forall má_náplň.Vegetariánská_náplň$.

Jinou záležitostí kupříkladu je, že množinu všech deklarovaných podtříd určité třídy nelze apriorně chápat jako rozklad této třídy, ani tyto podtřídy nejsou vzájemně disjunktní. To může vést k problémům např. ve spojení s další, pro databázisty nezvyklou, skutečností: omezení na definiční obor a obor hodnot relace se chápou jako logické formule (nad kterými probíhá odvozování), ne jako integritní omezení. Pokud tedy kupříkladu deklarujeme, že definiční obor relace $má_náplň$ je třída $Pizza$, a poté uvedeme třeba u třídy $Šlehačkový_dort$ (která nebude explicitně prohlášena za disjunktní s třídou $Pizza$) podmínku $\exists má_náplň.Šlehačka$, důsledkem bude, že odvozovací nástroj zařadí $Šlehačkový_dort$ jako podtřídu třídy $Pizza$.

Mnoho dalších „paradoxů“ je podrobně rozebráno v [38].

Logické návrhové vzory

Třetí skupinu problémů spojených s tvorbou ontologie ve formálním jazyce lze charakterizovat tak, že není na první pohled patrné jednoznačné převedení intuitivní konceptualizace do daného, omezeného jazyka. Pro řešení těchto problémů jsou postupně vytvářeny⁵⁶ návrhové vzory v mnoha směrech blízké vzorům známým v softwarovém inženýrství. Od obsahových vzorů uvedených v části 5.3 se liší vazbou na konkrétní jazyk a naopak (relativní) nezávislostí na věcné doméně.

Velmi frekventovaným problémem v OWL je například nemožnost přímo modelovat relace s *aritou* větší než 2. Návrhový vzor [31] doporučuje pro většinu případů provést reifikaci relace, tj. n-ární relace (např. vztah mezi prodávajícím, kupujícím, kupovaným předmětem, případně cenou a dalšími účastníky relace) je převedena na třídu (např. „Obchodní transakce“) a ostatní objekty jsou již pak k ní připojeny binárními relacemi.

Složitějším problémem, řešeným vzorem s mnoha alternativami [32], je situace, kdy potřebujeme na jedné straně modelovat hierarchickou strukturu (např. živočišných druhů, rodů, čeledí atd.) a na druhé straně na prvky hierarchie odkazovat relací (např. spojující určité informační zdroje se skupinami živočichů, o kterých pojednávají). Zdá se, že bychom tedy potřebovali, aby mohla samotná *třída* (např. „Lev“) a nikoliv jen její konkrétní instance (konkrétní lvi) být *argumentem relace*. To sice lze, ale pouze v OWL Full, nad kterým se odvozuje obtížněji, než nad OWL DL. Jendnou z alternativ je kupříkladu vytvořit k hierarchii tříd paralelní hierarchii individuí, vyjadřujících různé *tématiky* (např. „tématika lvů“); relace

⁵⁶ Nejčastěji v rámci pracovní skupiny W3C „Semantic Web Best Practices and Deployment“, viz <http://www.w3.org/2001/sw/BestPractices/>.

„pojdnává o“ pak může odkazovat na ně a nikoliv na třídy interpretované jako množiny fyzických živočichů.

Posledním problémem (a návrhovým vzorem), o kterém se zmíníme, je potřeba odlišného modelování proměnlivých aspektů objektů oproti trvalým. Příkladem je rozlišení tvrzení, že určitý objekt je člověk, od tvrzení, že tentýž objekt je učitel. Zatímco člověkem je objekt po celou dobu své existence (jde o esenciální charakteristiku, viz část 5.3), učitelem přestane být nejen tehdy, když zemře, ale i tehdy, když změní povolání nebo odejde do trvalého důchodu – z hlediska ontologie bychom pak museli tentýž objekt přemísťovat mezi třídami. Být učitelem je totiž *role*, kterou daný objekt hraje, a nikoliv třída v základním smyslu slova. Primitivním řešením problému je např. třídu *Učitel* vůbec nezavádět a spokojit se s relací *být_učitelem*. To však vede ke sníženým možnostem odvozování. [50] proto navrhuje modelovat role (např. při konverzi z některého jazyka, který má pojem role přímo ve svém metamodelu) v OWL pomocí komplexního logického vzoru. Vzor rozlišuje třídy odpovídající základním, konceptům jako je *Člověk* nebo *Škola*, třídy odpovídající rolím („role concepts“), jako je *Role_učitele*, a třídy odpovídající držitelům role („role holder“), jako je *Učitel*.

V rámci vývoje jazyka, jako je OWL, existuje přirozený trend k přesouvání frekventovaných konstrukcí z úrovně logických návrhových vzorů na úroveň rozšíření normy jazyka o potřebné konstrukce. Tak se například problém tzv. *kvalifikovaných kardinalitních omezení* (současný požadavek na kardinalitu relace a na třídy, jejíž instance smějí být hodnotami relace), původně nedokonale řešený pouze na úrovni návrhového vzoru, definitivně vyřešil zařazením potřebné konstrukce do nové verze jazyka⁵⁷, OWL 1.1.

7 Specializované nástroje pro tvorbu a využívání ontologií

7.1 Editory ontologií

Z hlediska ontologického inženýrství jsou samozřejmě základními nástroji editory, které umožňují ontologii vytvářet, ale zpravidla také testovat, a v mnoha případech jsou vybaveny i rozšiřujícími moduly s rozličnou funkcí.

Pro ontologie v OWL (a RDFS) je v současnosti jednoznačně nejrozšířenějším editorem volně šířený stanfordský *Protégé*⁵⁸ se svým OWL Plugin, který je sám složen z řady volitelných komponent. *Protégé* se svou stabilitou vyrovná komerčním nástrojům, a díky více než stovce zásuvných modulů (např. pro vizualizaci, tvorbu pravidel ve SWRL, mapování ontologií, validaci, importy/expoty) řeší alespoň jednoduchým způsobem prakticky všechny myslitelné úlohy v rámci ontologického inženýrství. Jednodušší (a tudíž méně časově a paměťově náročný) editor s důrazem na webové adresování a kombinování více ontologií je *SWOOP*⁵⁹, vzniklý na univerzitě v Marylandu. Mezi komerčními nástroji dnes patří k hlavním konkurentům

⁵⁷ <http://www.webont.org/owl/1.1/>

⁵⁸ <http://protege.stanford.edu/>

⁵⁹ <http://www.mindswap.org/2004/SWOOP/>

Tutoriál

*TopBraid Composer*⁶⁰ od firmy TopQuadrant, *OntoStudio* (nástupce *OntoEdit*) od firmy Ontoprise⁶¹, a *SemanticWorks* od firmy Altova⁶².

Tvorba *Topic Maps* je podporována zejména nástroji firmy Ontopia⁶³ jako je *OKS* (Ontopia Knowledge Suite).

7.2 Další specializované nástroje

Vzhledem k obrovskému počtu vývojových týmů, které v oblasti ontologického inženýrství působí, zde můžeme uvést jen velmi stručný výběr.

Vzhledem k rostoucímu počtu dostupných ontologií se stále častěji nabízí možnost nevytvářet ontologii „na zelené louce“, nýbrž využít některou z ontologií již na webu dostupných. Analogicky k nedávné historii webových katalogů vs. vyhledávačů mají za sebou již svůj zenit „pasivní“ knihovny ontologií typu *DAML Library*⁶⁴. Popularitu si získávají *indexovací a vyhledávací služby* jako je *Swoogle*⁶⁵, případně novější *OntoSelect*⁶⁶ či *Watson*⁶⁷; zpravidla umožňují vyhledávat jak ontologie, tak i jimi anotované dokumenty. Zatímco *Swoogle* svým pojetím silně připomíná vyhledávač Google (s rozšířenou verzí algoritmu PageRank, nazvanou *OntoRank*), *OntoSelect* klade důraz na možnost vícejazyčného vyhledávání, a *Watson* na automatické hodnocení kvality nalezených ontologií a na možnost jejich kombinování.

S vyhledáváním ontologií úzce souvisí jejich vzájemné *mapování*, které je často podmínkou integrovatelnosti nezávisle vzniklých aplikací. Hlavní webový rozcestník pro tuto oblast⁶⁸ uvádí více než dvacet nástrojů pro zcela či částečně automatické mapování ontologií; vesměs se ale jedná spíše o experimentální prototypy.

Ze specializovaných nástrojů pro učení ontologií z textů je třeba zmínit především *Text2Onto*⁶⁹ a *OntoGen*⁷⁰, v obou případech se jedná o volně šířené nástroje.

8 Nasazení ontologií v praxi

Množství aplikací ontologií nasazených dlouhodobě (tj. nejen např. po dobu trvání určitého výzkumného projektu spolufinancovaného EU) v praxi stále výrazně zaostává za objemem teoretického a aplikačního výzkumu. Přesto pozvolna roste zájem komerčních i neziskových organizací, který je asi nejlépe patrný na statistikách prakticky orientovaných konferencí o sémantických technologiích – americké *Semantic Technology Conference*⁷¹ i evropské *ESTC*⁷².

⁶⁰ <http://www.topbraidcomposer.com>

⁶¹ <http://www.ontoprise.com>

⁶² <http://www.altova.com>

⁶³ <http://www.ontopia.net>; firma byla nedávno odkoupena firmou Bouvet.

⁶⁴ <http://www.daml.org/ontologies/>

⁶⁵ <http://swoogle.umbc.edu/>

⁶⁶ <http://olp.dfki.de/ontoselect/>

⁶⁷ <http://watson.kmi.open.ac.uk>

⁶⁸ <http://www.ontologymatching.org>

⁶⁹ <http://ontoware.org/projects/text2onto/>

⁷⁰ <http://ontogen.ijs.si/>

⁷¹ <http://www.semantic-conference.com/>

Lze říci, že nejlépe si, z hlediska všeobecného přijetí, vedou rozsáhlé doménové ontologie s jednoduchou strukturou a bez ambicí na rozsáhlejší automatické odvozování (např. medicínské či produktové taxonomie), a také poměrně malé ontologie, které jsou však spojeny s velkým objemem dat distribuovaně uložených v rámci sémantického webu (např. FOAF). Jednou z možných „průlomových“ aplikací v prostředí velkých firem je připravované řešení managementu znalostí ve firmě Vodafone⁷³. Možnosti ontologií pro integraci informací a konkurenční zpravodajství jsou dlouhodobě zkoumány i dalšími telekomunikačními společnostmi. Mezi ně patří France Télécom, který se podílí na testování nástrojů sémantického webu, nebo British Telecom, jehož vývojové centrum přišlo s vlastním nástrojem pro konkurenční zpravodajství využívajícím ontologie – Data Foundry [39]. Mezi další místa opakovaného nasazení ontologií patří instituce státní správy, kulturní organizace (např. muzea), mediální agentury, průmyslové podniky (např. automobilky a výrobci letadel), firmy pracující v oblasti lidských zdrojů apod. Topic Maps se uplatňují např. jako základ dynamicky upravovaného informačního portálu norského ministerstva kultury⁷⁴. Samostatnou kapitolu by si zasloužily aplikace lingvistických ontologií (např. WordNetu) při zpracování přirozeného jazyka, a tzv. extrakčních ontologií při získávání znalostí z WWW [8].

Obecně je poměrně pomalý nástup ontologií jako „běžného“ technologického řešení do praxe spojen se specifiky znalostních technologií jako takových: formalizované znalosti jsou (na rozdíl od jednoduchých dat) obtížně převoditelné do jednodušné podoby, pro jejíž zpracování lze efektivně vyvinout hromadně šířený softwarový nástroj. Každý projekt zahrnující ontologie je proto do značné míry jedinečný a vyžaduje nasazení expertů s předchozími zkušenostmi jak s technologickým řešením tak i s věcnou oblastí. Na druhé straně však mají již vzniklé ontologie, při dodržení některých zásad naznačených i v tomto textu, solidní naděje na opakované a dlouhodobé využívání.

9 Závěr

Cílem tohoto doprovodného textu k tutoriálu bylo pomoci posluchačům i případným dalším čtenářům zlepšit si orientaci v rozsáhlém „světě“ ontologií, propojit si obvyklou představu o ontologii jako filosofické disciplíně s moderním informatickým pojetím, a získat představu o zdrojích a postupech, které jim mohou být užitečné, pokud by v určité aplikaci potřebovali využít existující ontologii nebo dokonce sami navrhnout ontologii vlastní.

10 Poděkování

Autoři by rádi poděkovali kolegům, kteří se podíleli na testování některých nástrojů pro práci s ontologiemi, zejména pak Janu Nemravovi, Ondřejovi Švábovi a Janu

⁷² <http://www.estc2007.com/>

⁷³ <http://www.w3.org/2001/sw/sweo/public/UseCases/Vodafone-es/>

⁷⁴ <http://www.ontopia.net/solutions/oks-applications.html>

Tutoriál

Zemánkovi. Užitečné jim byly také zkušenosti získané v rámci hostujících přednášek Giancarla Guizzardiho na pracovišti VŠE Praha⁷⁵ a spolupráce s partnery v projektech EU K-Space⁷⁶ a Knowledge Web⁷⁷.

Tutoriál je součástí přípravy kurzu „Znalosti a ontologické inženýrství“ v rámci nového magisterského studijního programu „Kognitivní informatika“ na VŠE Praha⁷⁸.

Literatura

1. Arndt, R., Troncy, R., Hardman, L., Staab, S., Schenk, S., Vacura, M., Nemrava, J., Šváb, O., Bailer, W., Petridis, K., Buitelaar, P.: *K-Space Deliverable 4.1*. Přístupné online na adrese <http://kspace.qmul.net:8080/kspace/deliverables/KS-D4.1-KU-MultimediaOntologyInfrastructurev1.0.pdf>
2. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F. (Eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press 2003.
3. Benjamins, R., Casanovas, P., Breuker, J., Gangemi, A. (Eds.): *Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*. Springer, 2005.
4. Borst, P., Benjamins, J., Wielinga, B., Akkermans, H.: An Application of Ontology Construction. In: *ECAI'96 Workshop on Ontological Engineering*, Budapest, 5-16, 1996.
5. Buitelaar, P., Cimiano, P., Magnini, B. (eds.), *Ontology Learning and Population*, IOS Press, 2005.
6. Cimiano, P., Eberhart, A., Hitzler, P., Oberle, D., Staab, S., Studer, R.: *The smartweb foundational ontology*. SmartWeb Project Report, 2004.
7. Decker, S., et al.: The Semantic Web – on the respective roles of XML and RDF. *IEEE Internet Computing*, 2000.
8. Embley, D. W., Tao, C., Liddle, S. W.: Automatically extracting ontologically specified data from HTML tables of unknown structure. In: *Proc. ER'02*, 322-337, London, UK, 2002. Springer-Verlag.
9. Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, A., Pazos-Sierra, J.: Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. *IEEE Intelligent Systems & their applications*, 1999.
10. Gangemi, A.: Ontology Design Patterns for Semantic Web Content. In: *Proc. of International Semantic Web Conference 2005*. 262-276.
11. Gangemi, A., Pisanelli, D.M., Steve, G.: Some Requirements and Experiences in Engineering Terminological Ontologies over the WWW. In: *Proceedings of the 1998 Knowledge Acquisition Workshop KAW 98*.

⁷⁵ Materiály ke kurzu jsou přístupné na webové adrese <http://guizzardi.panrepa.org>.

⁷⁶ Viz <http://www.kspace-noe.eu>.

⁷⁷ Viz <http://knowledgeweb.semanticweb.org>.

⁷⁸ Viz <http://kogninfo.panrepa.org>.

12. Gangemi, A. et al.: A Core Ontology of Fishery and its Use in the Fishery Ontology Service Project. In: *Proceedings of the EKAW'04 Workshop on Core Ontologies in Ontology Engineering*, online <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-118/>
13. Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O.: *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web*, 2nd edition. Springer, 2007.
14. Grau, B. C., Horrocks, I., Parsia, B., Patel-Schneider, P., Sattler, U.: Next Steps for OWL. In *Proc. of OWL: Experiences and Directions*, available electronically at CEUR, 2006.
15. Gruber, T. R.: A Translation Approach to Portable Ontologies. In: *Knowledge Acquisition*, 5(2), 1993, str. 199–220.
16. Gruber, T. R., Olsen, G. R.: An Ontology for Engineering Mathematics. *Principles of Knowledge Representation and Reasoning*, 1994.
17. Gruninger, M., Fox, M. S.: Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95, Montreal, 1995.
18. Göckel, R. *Lexicon philosophicum*. Frankfurt a.M. 1613.
19. Guarino, N.: Formal Ontology and Information Systems. in: Guarino, N. (ed) *Proceedings of FOIS'98*, IOS Press, Amsterdam: Trento, Italy, 1998, str. 3–15.
20. Guarino, N., Welty, C.: Evaluating ontological decisions with ONTOCLEAN. *Communications of the ACM*, 45(2):61--65, February 2002.
21. Haarslev, V., Moller, R.: RACER system description. In: *Proc. of IJCAR 2001*, 2001.
22. Heller, B., Herre, H.: Ontological Categories in GOL. *Axiomathes* 14(1):57-76 Kluwer Academic Publishers, 2004.
23. Hennig, W.: *Grundzüge einer Theorie der phylogenetischen Systematik*. Berlin 1950.
24. Lorhard, J. *Ogdoas scholastica*. St. Gallen. 1606.
25. Maedche, A.: *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, 2002.
26. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: *Ontology library*. WonderWeb Deliverable no.18, 2002. Online available from <http://wonderweb.semanticweb.org/deliverables/D18.shtml>.
27. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: *The WonderWeb Library of Foundational Ontologies (WFOL)*. Technical report, WonderWeb Deliverable no. 17, 2002. <http://wonderweb.semanticweb.org/deliverables/D17.shtml>, 2002.
28. McCarthy, J.: Circumscription — A Form of Nonmonotonic Reasoning. In: *Artificial Intelligence*, 13, 1980, str. 27–39.
29. Motta, E.: *Reusable Components for Knowledge Modelling: Principles and Case Studies in Parametric Design*. IOS Press, Amsterdam, 1999.

Tutorial

30. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.
31. Noy, N., Rector A.: *Defining N-ary Relations on the Semantic Web: Use With Individuals*. W3C Working Draft 21 July 2004 Latest version: <http://www.w3.org/TR/swbp-n-aryRelations>.
32. Noy, N., Uschold, M., Welty, C.: *Representing Classes As Property Values on the Semantic Web*. W3C Working Group Note 5 April 2005. Latest version: <http://www.w3.org/TR/swbp-classes-as-values/>.
33. Oberle, D., Lamparter, S., Grimm, S., Vrandečić, D., Staab, S., Gangemi, A.: Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems. *Journal of Applied Ontology*, 1(2):163–202, 2006.
34. Oltramari, A., Gangemi, A., Guarino, N., Masolo, C.: Restructuring WordNet's top-level: the OntoClean approach. In: *Proceedings of OntoLex2002: Workshop on Ontologies and Lexical Resources*, Las Palmas, Canary Islands, 2002.
35. Øhrstrøm, P., Andersen, J., Schärfe, H.: What Has Happened to Ontology. *Conceptual Structures: Common Semantics for Sharing Knowledge 2005*, 425-438.
36. Pan, J. Z.: *Description Logics – Reasoning Support for Semantic Web*. PhD Thesis. <http://dl-web.man.ac.uk/thesis.shtml>
37. Reale, G.: *Platón. Pokus o novou interpretaci*. Praha 2005.
38. Rector, A. L. et al.: OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. *Proc. EKAW 2004*, 63-81
39. Roberts, M., Zancani, L., Enweani, B.: On The Application of Semantic Technologies in Model-Driven Telecommunications OSS Systems. In: *Proc. ESTC'07*, Vienna, Austria.
40. Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Porzel, R., Zorn, H.-P., Micelli, V., Schmidt, C., Weiten, M., Burkhardt, F., Zhou, J.: *DOLCE ergo SUMO: On foundational and domain models in SWIntO (SmartWeb integrated ontology)*. Technical report, AIFB, University of Karlsruhe, 2006.
41. Schneider, L.: How to Build a Foundational Ontology. The Object-Centered High-level Reference Ontology OCHRE. In: Guenther, A., Kruse, R., Neumann, B. (eds), *KI 2003: Advances in Artificial Intelligence*. Proceedings of the 26th German Conference on Artificial Intelligence (KI 2003). LNCS 2821. Heidelberg: Springer, str. 120-134.
42. Schreiber, G. et al.: *Knowledge Engineering and Management – The CommonKADS Methodology*. MIT Press, Cambridge, Massachusetts; London, England, 1999.
43. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur A., Katz, Y.: Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2), 2007.
44. Smith, B.: *Ontology and Informations Systems*. SUNY at Buffalo. 2004.
45. Sowa, J. F.: *Conceptual Structures: Information Processing*. In: *Mind and Machine*, Addison-Wesley, Reading, MA, 1984.

Ontologické inženýrství

46. Sowa, J. F.: *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
47. Stoilos, G. et al.: Fuzzy OWL: Uncertainty and the Semantic Web. In: *Proc. Int'l Workshop OWL: Experiences and Directions, 2005*; <http://image.ntua.gr/papers/398.pdf>.
48. Strawson, P.F.: *Individuals: An Essay in Descriptive Metaphysics*, Routledge 1996.
49. Stubkjær, E.: Integrating ontologies: Assessing the use of the CyC ontology for cadastral applications. In: Bjørke, J.T. and H. Tveite, (Eds): *Proceedings of ScanGIS'2001*, Agricultural University of Norway, 2001, str. 171–184.
50. Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R.: A Model of Roles in Ontology Development Tool: Hozo. In: *Proc. EKAW 2006*, Springer-Verlag, LNCS 2006.
51. Svátek, V.: Ontologie a WWW. In: *Proc. Datakon 2002*, D. Chlapek (Ed.), Masarykova univerzita, Brno (2002), 27–55.
52. Svátek, V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: *7th Conference on Business Information Systems*, Poznaň 2004.
53. Tsarkov, D., Horrocks, I.: Efficient reasoning with range and domain constraints. In: *Proc. of the 2004 Description Logic Workshop (DL 2004)*, 2004.

Annotation:

Ontological engineering

The tutorial provides an introduction to ontological engineering, including the relationship to traditional ontology and to description logics reasoning. Attention is paid to the available support methods and tools, including structured methodologies, upper-level ontologies, design patterns and ontology learning tools. Practical applications of ontologies are also discussed.