Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky

Bakalářská práce

2007 **Daniel Kouba**

Zadání závěrečné bakalářské práce

Katedra:	**Informační a znalostní inženýrství**
Školní rok:	**2007/2008**
Jméno:	**Daniel Kouba**
Obor:	**Informatika**
Téma:	**Transformace mezi Topic Maps a RDF/OWL**
Zásady zpracování:	• Seznámit se s jazykem Topic Maps • Seznámit se s jazyky Topic Maps, RDF a OWL • Seznámit se s existujícími přístupy pro vzájemnou transformaci • Zhodnotit použitelnost existujících přístupů a případně navrhnout přístupy nové
Seznam odborné literatury:	• Antoniou, G. and van Harmelen, F. A *Semantic Web primer.* Cambridge, MA: MIT Press, 2004. 238 pp. ISBN 0-262-01210-3 • Hunting, S. *XML topic maps creating and using topic maps for the Web.* Boston: Addison-Wesley. (2003) • Garshol, L. M. *Living with topic maps and RDF.* (2003)
Vedoucí bakalářské práce:	**Doc. Ing. Vojtěch Svátek, Dr.**
Datum zadání bakalářské práce:	**Říjen 2007**
Termín odevzdání bakalářské práce:	**Prosinec 2008**

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

Fakulta informatiky a statistiky
Katedra informačního a znalostního inženýrství

Transformace mezi Topic Maps a RDF/OWL

Bakalářská práce

Daniel Kouba

Vedoucí práce: Doc. Ing. Vojtěch Svátek, Dr.

Listopad 2007

Abstrakt

Cílem této práce je nastínit princip transformace mezi RDF/OWL a Topic Maps (a obráceně) a zhodnotit použitelnost existujících nástrojů, které konverzi umožňují.

Tato práce navíc poskytuje stručný, ale komplexní přehled technologií RDF a Topic Maps. Názorně ukazuje princip obou modelů, shrnuje výhody a rozdíly mezi modely, principy možné transformace mezi modely a popis nástrojů, které konverzi umožňují. Rovněž se snaží zhodnotit kvalitu konverze a míru zachování informací ve výsledném modelu. Navíc se práce zabývá historií obou standardů, a snaží se nastínit i jejich budoucnost. Práce se lehce zabývá i dotazovacími jazyky pro oba formáty.

Práce je rozdělena na kapitoly, které se postupně zabývají klíčovými tématy a to RDF, Topic Maps, rozdíly mezi rodinami standardů, princip vzájemné konverze a nakonec kapitoly, která se týká samotných nástrojů pro konverzi. Princip vzájemné transformace je převzat od Garshol L. [8].

Přínosem této práce je přehledné shrnutí hlavních technologií, pojmů, rozdílů, faktů a zajímavostí v jednom dokumentu. Po přečtení prvních 4 kapitol by se měl v tomto oboru zorientovat i naprostý začátečník. V závěru práce jsou zhodnoceny nástroje pro konverzi mezi formáty.

Abstract

Objective of this document is to show the process of the conversion between RDF/OWL and Topic and vice versa, and then try to review existing tools, which can do it.

This paper provides brief, but complete summary of standard families. Clearly shows a data model of both RDF and Topic Maps, sums up advantages and disadvantages, introduces the history and future of these models, brings tools for a conversion. Moreover this work tries to evaluate accuracy and naturalness of conversion result. This work is based on Lars Marius Garshol's [7].

Contribution of this work is well-arranged summary of all technologies, terms, differences and facts about RDF and Topic Maps standards families in one document.

Poděkování

Velmi děkuji vedoucímu mé práce **doc. Ing. Vojtěchu Svátkovi Dr**., že mi umožnil pod jeho vedením zpracovat toto téma, které shledávám opravdu velice zajímavým. Rovněž bych chtěl poděkovat **Ing. Jiřímu Koskovi** za jeho velmi povedený a přehledný tutoriál k Topic Maps, který mě velmi inspiroval a za jeho celoživotní práci (příručky HTML, PHP), která mě přivedla k mému povolání, které mě tak baví. Samozřejmě děkuji oběma výše zmíněným pánům za vedení kurzu **4IZ440**.

Děkuji

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil pouze literaturu uvedenou v přiloženém seznamu. Nemám námitek proti půjčení práce se souhlasem katedry ani proti zveřejnění práce nebo její části.

V Praze dne 6. května 2005

Daniel Kouba

Obsah

1 Úvod ... 10
2 RDF ... 10
2.1 Historie RDF ... 11
2.2 Princip RDF ... 11
2.2.1 Omezení ... 12
2.2.2 Datový model ... 12
2.2.3 Syntaxe ... 12
2.2.4 Identifikace zdrojů ... 14
2.2.5 Dotazovací jazyky ... 14
2.3 OWL ... 15
2.4 Klady, zápory a využití RDF ... 16
2.4.1 Použití RDF ... 16
2.4.2 Klady ... 17
2.4.3 Zápory ... 17
3 Topic Maps ... 17
3.1 Historie TM ... 18
3.1.1 Budoucí vývoj Topic Maps ... 18
3.2 Princip TM ... 19
3.2.1 Omezení ... 19
3.2.2 Datový model ... 19
3.2.3 Základní elementy Topic Maps ... 19
3.2.4 Datové úrovně ... 20
3.2.5 Identifikace témat ... 21
3.2.6 Obory platnosti (scope) ... 21
3.2.7 Syntaxe ... 22
3.3 Dotazovací jazyky ... 24
3.4 Klady, zápory a využití TM ... 25
3.4.1 Použití TM ... 25
3.4.2 Klady ... 26
3.4.3 Zápory ... 26
4 Rozdíly mezi RDF a TM ... 26
4.1 Reprezentace věcí symboly ... 27
4.2 Výroky, přiřazování ... 27
4.3 Identita ... 29
4.4 Reifikace ... 29
4.5 Kvalifikace (Obory platnosti) ... 30

4.6 Typy a podtřídy ... 31
4.7 Slučování ... 31
5 Princip vzájemné transformace mezi technologiemi ... 31
5.1 Proč transformovat ... 31
5.2 Transformace dat z RDF do TM ... 32
5.2.1 Modelování TM v RDF ... 32
5.2.2 Tvorba mapování pro specifické slovníky ... 32
5.3 Transformace dat z TM do RDF ... 36
5.4 Transformace OWL ... 38
5.4.1 Shrnutí ... 40
5.5 Transformace RDF Schema ... 40
6 Existující nástroje pro konverzi ... 40
6.1 Nástroje pro konverzi TM do RDF ... 41
6.1.1 OKS (Ontopia knowledge Suite) – Ontopoly ... 41
6.1.2 TM4L (TM for E-Learning) ... 44
6.2 Nástroje pro konverzi RDF do TM. ... 44
6.2.1 TM4L ... 44
7 Závěr ... 47
8 Citovaná literatura ... 48
9 Seznam použitých zkratek ... 50
10 Přílohy ... 51

1 Úvod

Žijeme v informační době a potřeba informací stále stoupá, ovšem s množstvím informací vznikají problémy s jejich ukládáním, tříděním, vyhledáváním, ověřováním a především s jejich porozuměním. Proto vznikly nezávisle na sobě hned dva formáty pro prezentaci metadat a znalostí (RDF a Topic Maps[1]). Oba mají své nesporné výhody a jsou určené k trochu jinému použití, ovšem jsou si v mnohém podobné. Pro budoucí vývoj znalostního inženýrství bude myslím velmi podstatným milníkem překonání tohoto rozdílu, v ideálním případě sjednocení těchto technologií do jedné, která by se stala standardem pro práci se znalostmi.

Tématem této práce je tedy **transformace mezi TM a RDF/OWL** . Cílem této práce je shrnout principy, výhody a rozdíly mezi jednotlivými modely, principy možné transformace mezi modely a popis nástrojů, které toto umožňují. Rovněž se pokusím zhodnotit kvalitu konverze a míru zachování informací ve výsledném modelu. Pro toto téma jsem se rozhodl, protože mě velmi zaujal kurz **4IZ440 Reprezentace a zpracování znalostí na WWW[2] (**ze kterého čerpám) a **4IZ229 Úvod do umělé inteligence[3]**. Myslím, že zpracování znalostní není jen velmi zajímavý předmět, ale velmi významné rozvíjející se odvětví, které bude mít rozhodující vliv na vývoj webu, vyhledávání a poskytování informací, a neposlední řadě na vývoj umělé inteligence. Nová generace vyhledávacích enginů ozbrojená znalostmi v těchto formátech a vhodnými ontologiemi, by mohla poskytovat relevantní výsledky na dotazy typu *„Zobraz adresu pizzerie a MHD spojení, kde je nejlevnější pizza capricciosa, kilometr od mého bydliště.* “. Dokonce věřím, že jednou díky masovému rozšíření znalostní základny a vývoji ontologií, bude umělá inteligence schopná chápat význam jednotlivých výroků a schopna nezávisle tvořit RDF data např. z holého textu či rozhovoru. Metodický postup vypracování této práce má několik etap

1. Seznámení se s tématem
2. Sehnání literatury, vytvoření bibliografie
3. Vytvoření osnovy práce
4. Zpracování jednotlivých kapitol, přílohy a použitých zkratek.

Jediným omezením této práce, je nedostatek literatury o vzájemné transformaci, neboť se tímto problémem až dosud zabývalo jen pár odborníků. Má práce bude velmi úzce spojena s dokumentem [8], neboť se tohoto tématu bezprostředně týká. Předpokládá se, že čtenář této práce má základní znalosti o technologiích XML, HTML a WWW. V této práci se vyskytuje spousty termínů, které byly přeloženy z původních materiálů, proto má většina těchto pojmů v závorce uvedený původní anglický překlad, aby se člověk mohl lehce zorientovat i v originálních materiálech a vyvaroval se tak nepochopení jiných prací, kde je použito jiných termínů.

Nejprve se seznámíme s jednotlivými formáty, následně si vysvětlíme rozdíly mezi těmito systémy a ve finále popíšu princip a mechanismus transformace mezi formáty

2 RDF

RDF je zkratka pro **Resource Description Framework** , což je v překladu systém pro popis zdrojů. RDF je rodina **World Wide Web Consortium (W3C)**[4] specifikací původně navržených jako

[1] Topic Maps – dále jen TM
[2] Info o předmětu : http://pes.vse.cz/main.php?action=PRED_INFO&id_predmetu=4IZ440
[3] Info o předmětu: http://pes.vse.cz/main.php?action=PRED_INFO&id_predmetu=4IZ229
[4] http://www.w3.org/ , více v seznamu použitých zkratek kapitola (9)

model pro ukládání metadat. Ovšem v praxi se ukázalo, že se dá s výhodou použít jako model pro práci a předávání znalostí.

RDF model je založen na myšlence výroku (**statement**), které se skládá z takzvané **trojice**. Každá taková trojice se skládá ze **subjektu, predikátu a objektu** (česky předmět, vlastnost, objekt). Kde subjekt je zdroj nebo nositel vlastnosti a předmět je hodnota této vlastnosti (pokud je to literál). Předmět může být i jiný zdroj. Například ve výroku „Pizza Hawai obsahuje ananas “, je subjektem *Pizza Hawai*, predikátem *obsahuje* a objektem *ananas*. Takovýto výrok serializovaný do XML může vypadat takto:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:pizza="http://www.example.com/pizza/">
    <rdf:Description rdf:about="http://www.example.com/pizza#hawai">
        <pizza:obsahuje>ananas</pizza:obsahuje>
    </rdf:Description>
</rdf:RDF>
```

Tento mechanismus pro popis zdrojů byl oddělením **W3C Semantic Web activity** určen jako hlavní komponenta pro další evoluční stupeň **World Wide Webu (WWW)** a to takzvaného Sémantického webu, kde se budou moci informace (či znalosti chcete-li) zpracovávat, ukládat a předávat ve strojově srozumitelném formátu, což nám umožní používat a vyhledávat informace mnohem rychleji, efektivněji a především přesněji a s větší jistotou. Kromě této iniciativy W3C se začalo RDF hojně používat i ve znalostním inženýrství pro práci se znalostmi.[26]

2.1 Historie RDF

RDF není první model, který se zabýval popisováním zdrojů, RDF má hned několik předků. Technicky nejblíže bylo **MCF**[5], projekt zahájený panem Ramanathanem V. Guhou v Apple Computer a stále pokračuje s pomocí Tima Braye majitele Netscape Communications Corporation. Základní myšlenka byla převzata od Dublin Core Community a **PICS**[6] a byla základním kamenem pro vývoj RDF

W3C uveřejnilo specifikace pro RDF datový model s XML syntaxí jako doporučení v roce 1999. Tím začala práce na nové verzi, která vyšla v roce 2004 jako soubor souvisejících specifikací k původnímu RDF. Mezitímco existovalo pár implementací založených na specifikaci z r.1999, bylo třeba doporučení kompletně aktualizovat. Osvojení vylepšených specifikací bylo celosvětově přijato, dřív než vyšlo jejich plné veřejné předvedení, na rozdíl od většiny dřívějších technologií W3C. Většina nově příchozích vývojářů ani nemělo ponětí, že nějaká dřívější specifikace vůbec existovala.

MIME media typ ***application/rdf+xml*** byl registrován pod **RFC 3870**. Doporučuje RDF dokumentům použít nové specifikace. Navíc k serializaci RDF do XML přidalo W3C konsorcium i takzvanou **Notation 3** (známé spíše pod zkratkou **N3**), která umožňuje díky serializaci bez použití XML jednodušší modelování pro začátečníky. [28]

2.2 Princip RDF

Základní princip RDF byl popsán hned v úvodu této kapitoly, čili že RDF definuje vztah mezi zdroji, vlastnostmi a hodnotami. Nyní se na tento model podíváme podrobně.

Celý systém RDF má 3 vrstvy:

1. Omezení - OWL,RDF
2. Datový model - RDF

[5] Více o MCF - http://en.wikipedia.org/wiki/Meta_Content_Framework
[6] PICS (Platform for Internet Content Selection - dřívější systém označování obsahu webu od W3C)

3. Syntaxe - RDF/XML ,N3, N-Triples, NTRIX, TURTLE

2.2.1 Omezení

V této vrstvě se definují omezení datového modelu. Tyto omezení mohou být např. **ontologie**[7] či RDF **schémata**, které popisují vztahy mezi koncepty, nezbytné podmínky pro správnost a úplnost výsledného RDF či připravují vhodné schéma pro určitou oblast dat v RDF, která se dá s výhodou znovupoužít.

- **Ontologie – Web Ontology Language** (**OWL**) je jazyk definující a konkretizující Webové ontologie. OWLovská ontologie může obsahovat nejen popis tříd, ale i jejich vzájemné vztahy a instance. OWL je navržen pro použití aplikacemi, které potřebují pracovat s obsahem informací namísto pouhé prezentace informací lidem. Použití ontologií nám umožňuje větší strojovou interpretovatelnou webového obsahu podporujícího XML, RDF a RDF schémata. Tato výhoda je umožněna díky poskytnutí přídavného slovníku s formální sémantikou. OWL je založen na dřívějších jazycích **OIL** a **DAML+OIL**[8] a nyní je také doporučením W3C Ontology Working Group [31]. Tomuto jazyku je věnovaná samostatná kapitola 2.3.
- **RDF Schema (RDF-S)** je rozšiřitelný jazyk pro reprezentaci znalostí, poskytuje základní elementy pro popis ontologií, zamýšlený ke strukturování RDF zdrojů. Těmto schématům se také říká RDF slovníky. Hlavní RDFS komponenty jsou dostupné i ve více pokročilém (více vyjadřujícím) jazyku OWL. [27] Těmito základními elementy jsou **Property, Class, SubClassOf, SubPropertyOf, Domain** a **Range** (zajímavé Informace o RDFS najdete v [19] a [22]).

2.2.2 Datový model

Základní elementy RDF

- **Resource** (zdroj) jednoznačně identifikován URI. Může se vyskytovat jako *subjekt, predikát či objekt*.
- **Property** (vlastnost) vyskytuje se jako *predikát* a je současně zdrojem.
- **Value** (hodnota, literál) datatypy jsou v RDF převzaty z **XML Schema**. Hodnota se může vyskytovat pouze jako objekt výroku.
- **Statement** (výrok) – jedná se o trojici, *subjekt, predikát, objekt*. [22]

RDF jako orientovaný graf (Directed labelled graphs)

Celý RDF model lze chápat jako orientovaný jmenný graf. Jednotlivé subjekty a objekty jsou **uzly (Nodes)** grafu, a vztahy mezi nimi jsou orientované úsečky, které je spojují. Tvrzení „Pizza Hawai obsahuje ananas“ by vypadalo jako tento graf:

OBRÁZEK 1 - RDF GRAF, ZDROJ: W3C, RDF VALIDATOR

2.2.3 Syntaxe

K tomu abychom mohli RDF model **uložit do souboru,** a **vyměňovat soubory mezi aplikacemi** potřebujeme RDF serializovat (převést informaci do textové podoby). K tomu slouží syntaxe. RDF umožňuje samozřejmě více syntaxí : [22]

---

[7] Termín ontologie je vysvětlen v použitých zkratkách kapitola (9)

[8] DAML+OIL je předchůdce OWLu, více informací v [20] nebo http://www.daml.org/2001/03/daml+oil-index

RDF/XML

nejběžnější a zároveň W3C doporučovaná syntaxe používající značkovací jazyk XML. RDF/XML má mediální MIME typ `application/rdf+xml`, s příponou **.rdf.** [28]

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description
     rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

N3 (Notation 3)

MIME media typ `text/rdf+n3` a přípona je **.n3**. N3 je úsporná syntaxe nepoužívající XML, která byla navržena především pro jednoduchost a srozumitelnost člověku. Soubory notace 3 jsou vždy v UTF-8. S touto syntaxí přišli pánové **Tim Berners-Lee**[9] a Dan Connolly[24]

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn";
  dc:publisher "Wikipedia".
```

N-Triples

velice jednoduchá jednořádková syntaxe, její MIME media typ je `text/plain` s příponou souboru **.nt**. Tyto soubory musí být vždy v US-ASCII kódování. Jedná se o oficiální W3C syntaxi, jejím autorem je Dave Beckett. [2]

```
<http://www.w3.org/2001/sw/RDFCore/ntriples/><http://purl.org/dc/ele
ments/1.1/creator> "Dave Beckett" .

<http://www.w3.org/2001/sw/RDFCore/ntriples/><http://purl.org/dc/ele
ments/1.1/creator> "Art Barstow" .

<http://www.w3.org/2001/sw/RDFCore/ntriples/><http://purl.org/dc/ele
ments/1.1/publisher> <http://www.w3.org/> .
```

TURTLE (Terse RDF Triple Language)

je podmnožinou **N3**, určenou pouze pro prosté RDF grafy. Tuto syntaxi vynalezl Dave Beckett a nejedná se o oficiální syntaxi. Její media MIME typ je `application/x-turtle` a přípona souboru **.ttl** .[30]

[9] Vynálezce WWW, ředitel W3C - http://www.w3.org/People/Berners-Lee/

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [
    ex:fullname "Dave Beckett";
    ex:homePage <http://purl.org/net/dajobe/>
  ] .
```

NTRIX

O této notaci jsem zjistil pouze, že existuje. Zdroj této informace [22].

2.2.4 Identifikace zdrojů

Jako subjekt RDF výroku je zdroj, který může být pojmenovaný buď **URI (Uniform Resource Identifier)**[10], nebo může být nepojmenovaný, takovým zdrojům se říká **Blank Nodes** (Prázdný uzel). Predikát je také zdroj reprezentující vlastnost. A jak jsme si již řekli, objektem může být buď další zdroj nebo **unicode string literal** (textová hodnota v unicodu).

URI má většinou podobu URL[11] adresy webových dokumentů. Obecně však není URI limitováno pouze na popis internetových zdrojů. Ve skutečnosti názvy zdrojů nemusí být vůbec dosažitelné, URI začínající jako URL adresa tedy *„http://…“*, použité jako subjekt výroku nemusí reprezentovat žádný zdroj přístupný přes internet např.: *http://www.recepty.cz/pizza#hawai*.

Je tedy nutné, aby se uživatele a tvůrci RDF výroků domluvili na jednotné sémantice resource identifikátorů. Například obecně používané RDF slovníky jako **Dublin Core Metadata**[12] jsou mapovány do URL prostoru např.: *http://purl.org/dc/elements/1.1/language*.[28]

2.2.5 Dotazovací jazyky

Dotazovací jazyky pro RDF fungují na stejném principu jako dotazovací jazyky relačních databází, tedy podle určitého dotazu vyberou určité hodnoty či podgraf RDF grafu. Dotazovacích jazyků pro RDF je více a se stále vyvíjí, ale standardem (doporučením W3C) je v současné chvíli jazyk **SPARQL**. [21] [24]

SPARQL (SPARQL Protocol and RDF Query Language)

SPARQL je dotazovací jazyk, který je standardizován DAWG[13]. SPARQL vyšlo jako předběžné doporučení v dubnu 2006, ale díky některým stále otevřeným problémům byl vrácen do dalšího vývoje[29]. **13. listopadu 2007** (měsíc zpět ve chvíli kdy píšu tuto práci) byl SPAQL publikovaný jako doporučení W3C pro dotazovací jazyk, XML formát pro výstup dotazu a protokol pro RDF[24]. Příklad dotazu, který vypíše všechny africké hlavní města afrických států[29].

[10] URI (Unifiorm Resource Identifier) – V překladu Jednotný identifikátor zdroje.

[11] URL (Uniform Resource Locator) – Určuje jednoznačné umístění dokumentu na internetu či intranetu a je podmnožinou URI.

[12] DC (Dublin Core) – slovník RDF URI predikátů o metadatech. http://dublincore.org/documents/dces/

[13] DAWG (RDF Data Access Working Group) – oddělení W3C konsorcia.

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y.
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa.
}
```

RQL (RDF Query Language)

RQL je dotazovací jazyk pro RDF a RDFS. RQL je založeno na funkčním přístupu OQL. Na stránkách [11] naleznete zajímavou on-line demonstraci, kde si můžete vyzkoušet jak RQL funguje nad RDF daty muzea. RQL je podle jejích autorů jednoduchý dotaz. Jazyk, který se dá naučit během jednoho dne. Příklad ukazuje dotaz na všechny zdroje, které byly modifikovány po roce 2000.[11]

```
select  X, Y from  {X}last_modified{Y} where Y >= 2000-01-01
```

RDQL

RDQL je dotazovací jazyk pro RDF pracující v prostředí **Jena**[14]. Myšlenkou RDQL je poskytovat datově orientovaný dotazovací model, který by umožňoval více popisující přístup (v originále „declarative approach“), který by doplnil procedurální **Jena API**. RDQL vychází ze staršího *rdfDB*, které umělo pracovat pouze s trojicemi bez schémat či ontologií. RDQL je dnes již zastaralé a bylo nahrazeno novějším a sofistikovanějším jazykem **SPARQL**. Příklad ukazuje dotaz, který vypíše všechny křestní jména všech Shakespearů.[18]

```
SELECT ?krestniJmeno
WHERE (?y  <http://www.w3.org/2001/vcard-rdf/3.0#Family>
     "Shakespeare") ,
         (?y  <http://www.w3.org/2001/vcard-rdf/3.0#Given>
     ?krestniJmeno)
```

SeRQL (Sesame RDF query Language)

Je RDF/RDFS dotazovací jazyk, který je v současnosti vyvíjí **Aduna** jako část **Sesame**. Snaží se vhodně kombinovat nejlepší prvky ostatních dotazovacích jazyků jako RQL, RDQL, N-Triples a N3 a přidává nějaké vlastní. Nejvýznamnější možnosti SeRQL jsou podpora RDFS a XML datatyp, transformace grafu, syntaxe **path expressio**n a dobrovolné **path matching.**[19]. Příklad ukazuje dotaz, který najde všechny práce, které se týkají RDF a dotazování.

```
SELECT
   Author, Paper
FROM
   {Paper} rdf:type {foo:Paper};
           ex:keyword {"RDF", "Querying"};
           dc:author {Author}
USING NAMESPACE
   dc = <http://purl.org/dc/elements/1.0/>,
   ex = <http://example.org/things#>
```

Další dotazovací jazyky pro RDF

Další dotazovací jazyky pro RDF jsou *Versa, N3QL, Triple, RDQ, DQL, R-Device, RDFQ, rdfDB*.

2.3 OWL

[14] Jena je RDF API postavené na programovacím jazyce Java

S tímto ontologickým jazykem jsme se seznámili už v kapitole 2.2.1, nyní se na něj podíváme trochu blíže. OWL je RDF slovník, ale popisovat jej jako jazyk pro omezení je trochu nekorektní, protože tento jazyk jde mnohem dál. OWL má schopnost omezovat hodnoty vlastností a navíc podporuje omezení kardinality (početnosti). Ale OWL jde ovšem ještě dál, umožňuje dělat něco, co se dá nejlépe nazvat sémantickou anotací RDF dat. To umožňuje některé prvky interpretace RDF popsat, proto se nejedná ani tak o omezovací jazyk, ale ontologii. OWL je určený pro podporu strojového zpracovávání informací, poskytuje informace o vztahu mezi koncepty a význam jednotlivých pojmů v ontologii. [31]

OWL se skládá ze tří podmnožin s různou vyjadřovací schopností určené pro různé komunity implementátorů a uživatelů.

- **OWL Lite** je podmnožinou slovníku OWL určená pro ty, kteří potřebují jednoduchý jazyk, který se lehce implementuje. Většina schopností pro sémantickou anotaci a pár možností omezení je vynechána.
- **OWL DL (Description Logics)** obsahuje celý slovník, ale omezuje, jak může být použit, kvůli poskytnutí lepší optimalizace pro stroje sloužící k logickému odvozování.
- **OWL Full** je pro uživatele, kteří potřebují maximální vyjadřovací schopnost a zároveň nechtějí být syntakticky omezováni a nevadí jim, že nemají žádné záruky pro výpočetní úlohy jako např. odvozování. [25]

O jednotlivých vlastnostech jazyku OWL bude řeč v kapitole 5.4.

2.4 Klady, zápory a využití RDF

Nyní se podíváme na využití RDF v praxi, na jeho silné a slabé stránky a na jeho konkrétní implementace

2.4.1 Použití RDF

Podívejme se na uplatnění RDF.

Aplikace pro metadata

RDF se dá s výhodou použít v těchto oblastech: [22]

- Katalogizace
- Vyhledávání zdrojů
- E-Commerce
- Hodnocení obsahu
- Znalostní inženýrství

Konkrétní aplikace RDF

A pokud Vás zajímá RDF v praxi, mohli by Vás zajímat níže uvedené aplikace.

- RDF Site Summary (**RSS**)- technologie pro publikování změn na webových stránkách, blozích. Systém distribuce článků. http://www.rssboard.org/rss-specification
- **FOAF** (Friend of a Friend) – RDF navržené k popsání lidí, jejich koníčků a jejich vzájemných vztahů.
- **DOAC** (Description of a Career),**DOAP** (Description of Project) – umožní sdílet přídavné informace k FOAFu
- http://musicbrainz.org/ - Portál poskytující informace o autorech, albech a skladbách uložené v RDF
- http://e-culture.multimedian.nl/demo/search - Portál, který poskytuje informace o malířích a jejich obrazech uložené v RDF

2.4.2 Klady

Největším Kladem RDF je jeho **jednoduchost** (subjekt, predikát, objekt), která jej činí pro člověka srozumitelným i v serializované podobě (v notacích N3,N-triples a Turtle). Další nespornou výhodou je, je může být přímo součástí X-HTML kódu vašich stránek. Nad RDF daty se poměrně snadno odvozovat nové vztahy.

2.4.3 Zápory

RDF je nejčastěji kritizováno za jeho přílišnou datovou velikost při serializaci (užvaněnost). Další velký problém RDF je, že trojice (subjekt, objekt, predikát) nemá dostatečnou vyjadřovací schopnost. Např. není možné vytvořit n-ární relace (vztah mezi více zdroji). Neexistuje jednoznačný a jednoduchý způsob jak vyjádřit tvrzení o tvrzení, takzvanou reifikaci[15].

3 Topic Maps

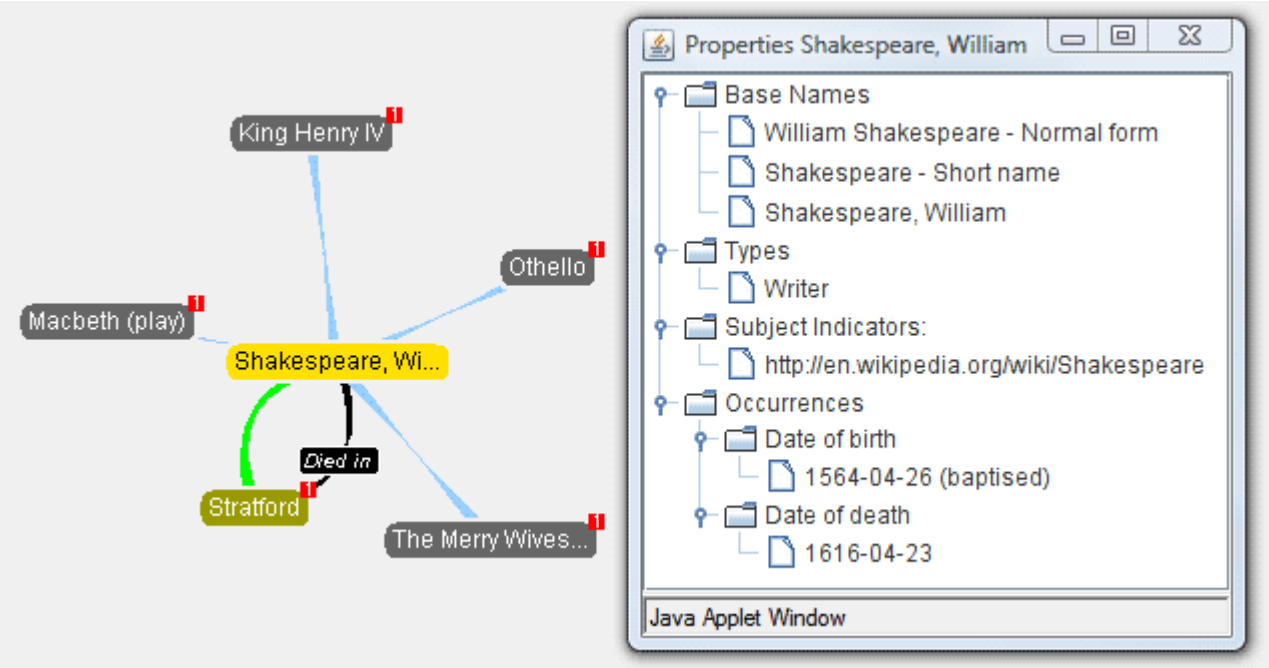
Topic Maps (Mapy témat) jsou mezinárodní norma **ISO**[16] pro reprezentaci a sdílení znalostí. Mapy témat jsou generalizace původní myšlenky rejstříků na konci knihy. TM byly navrženy na rozdíl od RDF pro reprezentaci znalostí určená především pro lidi. Obě tyto technologie jsou sice podobné, ale v některých důležitých aspektech se liší, což velmi znesnadňuje jejich vzájemnou integraci. O těchto rozdílech bude řeč v kapitole 4. [12]

Základní myšlenkou TM je **téma (Topic)**, které zastupuje nějaký předmět z reálného světa. Toto téma je popisováno použitím **vztahů a výskytů**. Tyto vzájemné **vztahy (Associations)** mezi tématy a jejich **výskyty (Ocurrences)** tvoří mapu témat. Výskyty charakterizují vlastnosti jednotlivých témat např.: název, identifikátor, popis. A vztahy popisují (jak jinak) vztahy mezi jednotlivými tématy. Pro usnadnění zapamatování základních elementů topic map slouží takzvané **TAO** (**T**opic **A**ssociation **O**ccurence). Mapy témat lze stejně jako RDF chápat jako graf. Viz obrázek: Témata jsou *Shakespeare ,Othello a Stratford* zobrazené jako uzly, vztahy mezi nimi jsou znázorněny jako spojnice a definují vztahy mezi tématy (*Shakespeare zemřel v Stratfordu*) . Vlastnosti tématu *Shakespeare* (Názvy,Identifikátor,Datum narození a umrtí) nejsou z grafu patrné, jelikož grafy se zde tvoří pouze na úrovni znalostní úrovně, kterou tvoří pouze témata a vztahy[17].

[15] Viz kapitola (4.4)

[16] TM – mezinárodní norma ISO/IEC 13250:2003 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38068

[17] Více v kapitole 3.2 Princip TM (KOSEK,J na svých stránkách [12] tyto úrovně nazývá vrstvy, ale já jsem si dovolil záměrně použít jiné označení, aby nedošlo k nedorozumění v mé terminologii)



OBRÁZEK 2 - TM GRAF, ZRDOJ: ONTOPIA OMNIGATOR, MAPA TÉMAT OPERA.LTM
HTTP://WWW.ONTOPIA.NET/OMNIGATOR/PLUGINS/VIZ/VIZ.JSP?TM=OPERA.LTM&ID=SHAKESPEARE

TM mají složitější datový model, který umožňuje jednoduše používat takové hi-end features jako *jsou reifikace, n-ární vztahy, obory pravdivosti (scope) a slučování map*. O těchto pokročilých aspektech TM bude řeč v kapitole 4. Ale i přes tento složitější model jsou pro člověka mnohem přehlednější a intuitivnější nežli RDF. Já osobně jsem si je velmi oblíbil pro jejich přehlednost a propracovanost. Pro TM lze díky jejich datovému modelu vytvořit obecný prohlížeč. Za velice povedený považuji prohlížeč **Omnigator**[18] společnosti **Ontopia**, která pro mapy témat vytvořila kompletní framework **OKS (Ontopia knowledge Suite)** a je jedním z největších současných protagonistů této platformy.

3.1 Historie TM

Historie Topic Maps sahá do začátku 90let. Kdy DEC a O'Reilly sdílejí unixovou dokumentaci a chtějí vytvořit unifikovaný rejstřík, využitím **HyTime**[19]. V roce 1996 byl další vývoj TM přenesen na půdu **ISO** (International Organization for standardization) pod hlavičkou konsorcia **OASIS**[20]. V roce 2000 oficiálně vychází standard *ISO/IEC 13250:2000 Topic Maps*, se základním datovým modelem TM založeném na syntaxi HyTime a SGML, známým jako **HyTm** (HyTime Topic Maps). Časem se ukázalo za nezbytné použít **XML** pro lepší integraci s webem, což vyústilo v roce 2000 k vytvoření formátu **XTM**[21] . Konečně v roce 2003 vychází revize standardu *ISO/IEC 13250:2003 Topic Maps,* který již zahrnuje i formát XTM.[12]

3.1.1 Budoucí vývoj Topic Maps

V současné době se pracuje na revizi standardu *ISO 13250*, který bude obsahovat formální definici datového modelu. A dále se připravují standardy:

- pro dotazovací jazyk **TMQL** (TM Query Language[22] – ISO 18048) a
- jazyk pro omezení TM (Ontologie **TMCL**[23] –TM constraint language ISO 19756).

[18] Omnigator – součást komerčního balíčku OKS - http://www.ontopia.net/omnigator/
[19] HyTime - Více v kapitole 9 seznam použitých zkratek
[20] Oasis je neziskové konsorcium pro vývoj standardů http://www.oasis-open.org/
[21] XTM – více v kapitole 3.2.7**Chyba! Nenalezen zdroj odkazů.** Formáty zápisu TM
[22] Více o vývoji TMQL dotazovacího jazyku http://www.isotopicmaps.org/tmql/

3.2 Princip TM

Základní princip jsem uvedl hned v úvodu této kapitoly, detailnější popis témat, vztahů a výskytů popíšu v datovém modelu TM. Topic Maps mají podobně jako RDF 3 úrovně:

1. Omezení - TMCL,OSL,AsTMa!
2. Datový model - Topic Maps
3. Syntaxe - XTM,LTM,HyTM,AsTMa=,CTM

3.2.1 Omezení

V této vrstvě se definují stejně jako u RDF omezení datového modelu. U Topic Maps nehrají ontologie tak velkou roli jako u RDF, protože bylo navrženo především pro lidi, čili v TM nejsou potřeba složitější logické axiomy ontologií, které pro stroje nahrazují to, co si o doménových konceptech člověk přečte přímo v textu. S tímto faktem jde ruku v ruce, že z TM nedají odvozovat jiné zajímavé vztahy jako například v RDF. Omezení TM jsou důležité pro ověřování zda-li jsou informace v TM konzistentní a smysluplné (např. zdali se *nenarodil Shakespeare v Othellovi*, či jestli není *Macbeth město*). Poskytují nám pohled na strukturu map témat, takže aplikace pracující s těmito mapami mohou mít přehlednější prostředí a umožňují efektivnější navigaci v TM.[13]

- **TMCL – TM Constraint Language** je jazyk, který je právě vyvíjen pod hlavičkou projektu *ISO N458* a vyjde jako standard *ISO 19756*. Má být velmi podobný jazyku *OSL*, ale bude mocnější a používat jinou syntaxi.[13]
- **OSL – Ontopia Schema Language** je jazyk, který umožňuje popsat omezení v mapě témat. Tento jazyk funkcí velmi podobný RDFS. Umožňuje definovat třídy témat, výskytů, vztahů, rolí (vztahu) a názvů témat. Tento jazyk používá Ontopia ve svém editoru TM **Ontopoly**[24] (součást balíku OKS), který musím velmi pochválit, protože je to naprosto intuitivní a velmi efektivní nástroj**.** Schémata se ukládají do souborů .osl, nebo přímo do souboru s TM. Více podnětných informací najdete v tutoriálu[13].
- **AsTMa![25] –** Okrajový jazyk pro omezení TM, jedná se o iniciativu pana Roberta Barty. Vzhledem k plánovanému standardu TMCL a hojně používanému OSL nepředpokládám, že by tento jazyk pro omezení v budoucnosti sehrál významnější roli, proto se pouze zmiňuji o jeho existenci.

3.2.2 Datový model

Datový model Topic Maps je mnohem složitější než u RDF. K pochopení datového modelu TM je třeba pochopit jeho základní stavební kameny (elementy), datové úrovně, způsob identifikace témat a obory pravdivosti.

3.2.3 Základní elementy Topic Maps

Datový model tvoří především základní elementy TM (témata, vztahy, výskyty), kde každý z těchto elementů může mít svůj typ. Tyto typy jsou rovněž témata ve stejné mapě témat. Každá mapa témat obsahuje svoji ontologii.[12]

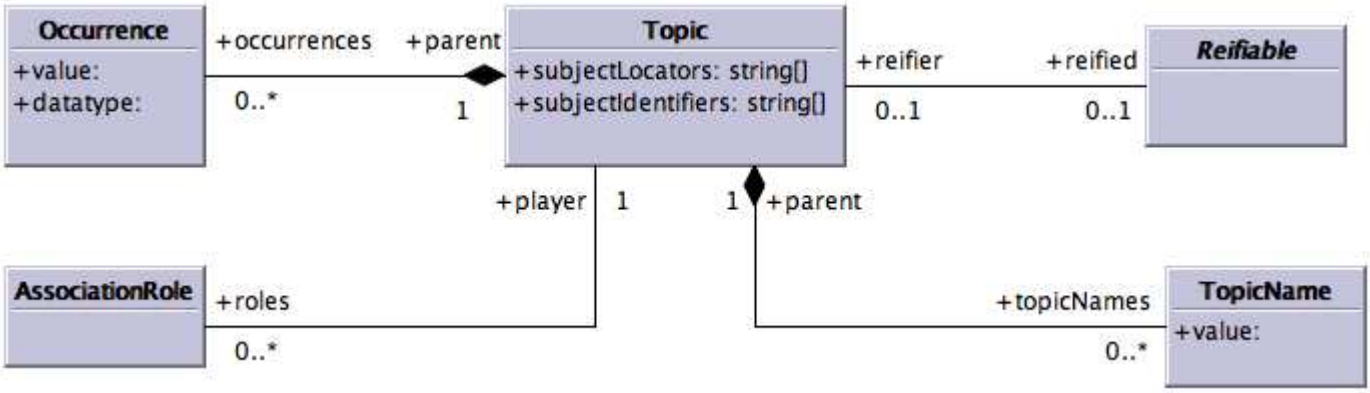
Základní elementy TM

- **Téma (Topic)** – Téma reprezentuje předmět z reálného světa nebo informační zdroj. Každé téma má tzv. **identifikátory** a charakteristiky. Tyto charakteristiky **tvoří názvy, výskyty a role ve vztazích**.
    - ***Název*** je v podstatě zvláštní případ výskytu – tedy zvláštní tím, že téma reprezentuje navenek. Každé téma může mít libovolný počet názvů. Tyto

[23] Více o vývoji TMCL na URL http://www.isotopicmaps.org/tmcl/
[24] Ontopoly – nástroj pro editaci TM - http://www.ontopia.net/ontopoly/
[25] AsTMa! – více informací v tutoriálu http://astma.it.bond.edu.au/astma!-tutorial.dbk?section=1

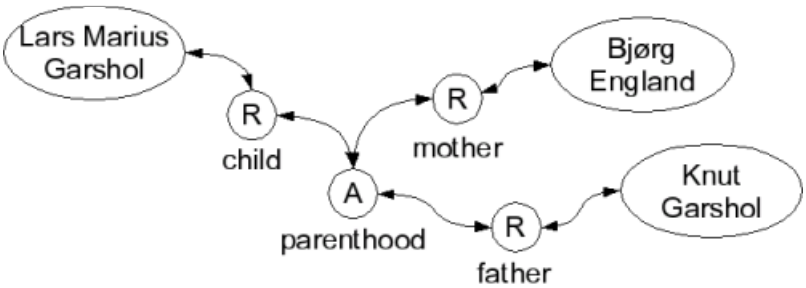
názvy nemusí být jednoznačné, většinou ani nemohou (synonyma, homonyma …) a tudíž neidentifikují téma. Tyto názvy mohou být např. názvy v různých jazycích.

- ***Výskyty*** viz níže.
- ***Role ve vztazích*** viz níže.
- ***Identifikátory*** slouží k identifikování tématu, jsou tedy na rozdíl od názvu unikátní a každé téma může mít libovolný počet identifikátorů. Identifikátory mají více typů, o nichž si povíme více v kapitole 3.2.5



OBRÁZEK 3 CHARAKTERISTIKY TEMA, ZDROJ HTTP://WWW.ISOTOPICMAPS.ORG/SAM/SAM-MODEL/#D0E1029

- **Výskyty (Ocurrence)** jsou obecné vlastnosti tématu. Každé téma může mít více výskytu nebo žádné. Výskyty mohou být buď **vnitřní (interní)** nebo **vnější (externí)**. Vnitřní výskyty jsou datové hodnoty uložené přímo v mapě témat (např.: datum narození), vnější výskyty jsou externí zdroje (dokumenty) v mapě témat zpravidla reprezentované jako URI.
- **Vztah (Asocciation)** - vyjadřují vztahy mezi jednotlivými tématy. Tyto vztahy nejsou orientované jako v RDF, ale orientace vztahu je dána **rolemi (roles)**. Specialitou Topic Maps je libovolná počet subjektů vztahu. Takže není problém např. definovat vztah rodičovství, kde jsou 3 role (takzvaný ternární vztah) a to *matka, otec, potomek* jak ukazuje obrázek. Rolí může být ve vztahu samozřejmě i více, takovým vztahům se říká N-ární vztahy.



OBRÁZEK 4 -- TERNÁRNÍ VZTAH, ZDROJ [7] HTTP://WWW.ONTOPIA.NET/TOPICMAPS/MATERIALS/TMRDF.HTML#N349

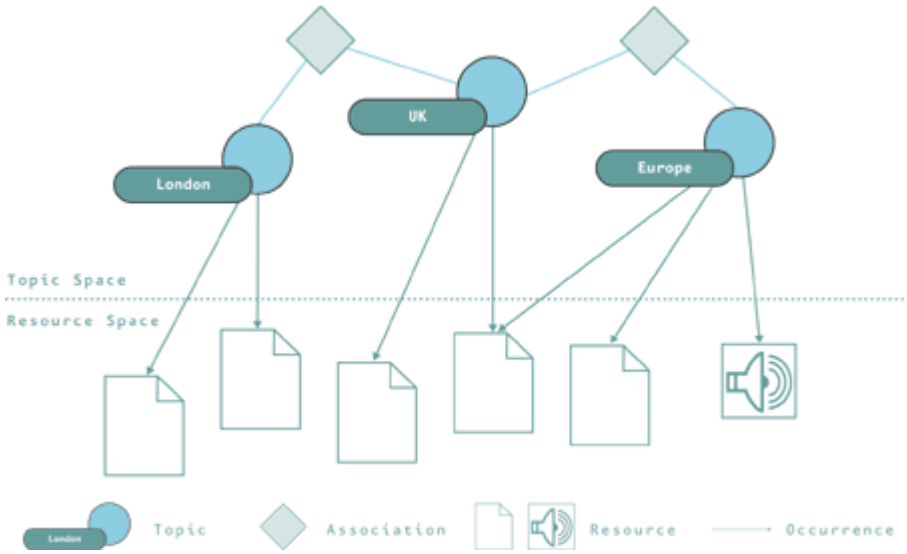
3.2.4 Datové úrovně

TM mají na rozdíl od RDF dvě datové úrovně. Tyto datové úrovně, které pan KOSEK, J nazývá ve svém tutoriálu[12] vrstvami je zde zároveň na stranách 4-7 i výborně ilustruje. Vřele doporučuji pro lepší názornost.

- **Úroveň znalostní (Topic Space)** – na této úrovni se pracuje se znalostmi. Tato úroveň popisuje vztahy mezi tématy.

- **Úroveň informační (Resource Space)** – na této úrovni se „vyskytuje“ samotný obsah a informační zdroje.

Pokud bychom TM chápali tak, jak byly původně zamýšleny, tedy jako rejstřík na konci knihy pak bychom mapu témat mohli chápat jako knihu. Informační úroveň by byla samotný obsah knihy a znalostní úroveň by byl věcný a jmenný rejstřík. Tyto vrstvy se v mapě témat propojují pomocí výskytů. *„Výskyt zachycuje vztah mezi tématem a souvisejícím informačním zdrojem“*[12].



OBRÁZEK 5 - PROPOJENÍ UROVNÍ, ZDROJ: SOFT.COM/EN-US/LIBRARY/AA480048.ASPX

3.2.5 Identifikace témat

Další specialita TM je systém identifikátorů tématu. V TM se vyskytují dva druhy **identifikátorů tématu**. TM na rozdíl od RDF rozlišuje, zdali je předmět, (který téma reprezentuje) informační zdroj nebo skutečný předmět. Pokud téma reprezentuje informační zdroj, pro identifikaci v mapě témat se používá tzv. **lokátoru předmětu (Subject locator)**. Tento identifikátor má zpravidla podobu URI (stejně jako u RDF). V opačném případě, tedy když téma reprezentuje předmět z reálného světa, se používá k určení **identifikátor předmětu (Subject identificator)**. U takovýchto předmětů nejde vyjádřit jejich identita přímo. K určení takového objektu lze ovšem použít informační zdroj (který už identifikovat jde), kterému se říká **indikátor předmětu**[12]. K identifikaci předmětu se pak používá identifikátor indikátoru. S výhodou lze použít odkazy na Wikipedii.

Například jako identifikátor spisovatele W. Shakespeara lze použít URI jakékoliv stránky, obsahující biograficky údaje o W. Shakespearovi a to např.

```
http://cs.wikipedia.org/wiki/William_Shakespeare
```

či

```
http://www.spisovatele.cz/william-shakespeare
```

Pro sdílení a slučování map je stejně jako u RDF třeba aby se autoři dohodli na jednotných identifikátorech map. TM je v tomto ohledu opět trochu dále než RDF, protože využívají systém **PSI** (Published Subject Indicator), tedy zveřejněných indikátorů předmětu. Doporučení pro vytváření sad PSI vydává konsorcium OASIS, pod názvem **OASIS Published Subject TC**[26][12].

3.2.6 Obory platnosti (scope)

[26] Existující sady PSI vytvořené v rámci OASIS naleznete na http://psi.oasis-open.org/

Každé téma má své charakteristiky, které platí jen v určitém kontextu. Jediné charakteristiky, které platí vždy, jsou identifikátory témat. Tyto oblasti platnosti se dají s výhodou použít při filtrování map. Obor platnosti je definován množinou témat[12]. Např.: Můžu mít mapu témat *opera.ltm*[27], která obsahuje vícejazyčné informace o operách, kde prohlížeč TM dokáže zobrazovat názvy témat v jazyce, ve kterém hovořím. Tyto obory platnosti jde také používat pro určení pravdivosti tvrzení, či autora nebo zdroje tohoto tvrzení. Každá informace může platit pouze v kontextu, kdy je autorem výroku určitý člověk, kterému přikládáme určitou důvěryhodnost. Základní obor pravdivosti je `unconstrained scope`(neomezený rozsah), kterým je myšleno, že neexistují limity pro platnost výroku (tedy, že platí vždy).

3.2.7 Syntaxe

Pro uložení datového modelu TM, je třeba obsah map serializozat (stejně jako u RDF). K tomu slouží syntaxe. Existuje mnoho syntaxí, ale vzhledem k tomu, že valná většina nástrojů pro práci s TM podporuje pouze formáty LTM a standardizované XML, nevidím jako významné zabývat se těmito okrajovými notacemi. Proto rozeberu podrobně pouze první dvě a o ostatní uvedu jen pro obecný přehled.

LTM (Linear TM notation)

LTM je jednoduchý textový formát pro TM. LTM se dá díky jednoduchosti a kompaktnosti psát v jakémkoliv textovém editoru. Přestože existuje spousta TM editorů, je stále třeba jednoduchý textový formát, který může být použit pro stručné a jasné vyjádření TM v emailech, diskuzích a podobných aplikacích. LTM navíc umožňuje rychle tvořit a upravovat malé TM pro osobní nebo demonstrační účely. Tato notace byla vyvinuta společností Ontopia a jedná se o stěžejní formát používaný v OKS, samozřejmě jde v tomto frameworku konvertovat LTM do standardizovaného XTM.[9]

[27] Tato TM je dostupná v OKS demo - http://www.ontopia.net/omnigator/

```
/* topic types */
[format       = "Format"]
[standard     = "Standard"]
[organization = "Organization"]

/* association types */
[format-for = "Format for"]
[defined-by = "Defined by"]

/* occurrence types */
[specification = "Specification"]
[homepage      = "Home page"]

/* topics, associations and occurrences */
[topic-maps : standard  = "Topic maps"
                        = "ISO/IEC 13250 Topic Maps" / fullname]
{topic-maps, specification,
   "http://www.y12.doe.gov/sgml/sc34/document/0129.pdf"}

[xtm : format = "XTM Syntax"]

[ltm : format = "The linear topic map notation";
                "linear topic map notation, the"
     @"http://www.ontopia.net/topicmaps/ltm-tech-report.html"]
{ltm, specification, "http://www.ontopia.net/topicmaps/ltm-tech-
report.html"}

format-for(ltm, topic-maps)
format-for(xtm, topic-maps)

defined-by(ltm, ontopia)
defined-by(xtm, topicmaps.org)

[ontopia : organization = "Ontopia AS"]
{ontopia, homepage, "http://www.ontopia.net"}

[topicmaps.org  : organization = "TopicMaps.Org"]
{topicmaps.org, homepage, "http://www.topicmaps.org"}
```

XTM (XML Topic Maps)

XTM bylo vytvořeno v roce 2000 v rámci iniciativy společnosti TopicMaps.org. Letos (2007) by měla (Nevím, jak to chtějí stihnout) vyjít nová revize standardu ISO/IEC 13250:2007, která bude obsahovat i standardizaci XTM. Jak je asi na první pohled zřejmé XTM je jak založeno na značkovacím jazyce XML. Jedná se o nejpoužívanější syntaxi.[23]

```
<topicMap>
    <topic id="t36">
      <baseName>
        <baseNameString>Hamlet</baseNameString>
      </baseName>
      <baseName>
        <baseNameString>The Prince Of Denmark</baseNameString>
      </baseName>
      <occurrence>
        <resourceRef
xlink:href="http://www.topicmaps.org/examples/hamlet.html" />
      </occurrence>
      <occurrence>
        <resourceRef
xlink:href="http://www.topicmaps.org/examples/hamlet_again.html" />
      </occurrence>
    </topic>

    <association>
      <member>
        <roleSpec><topicRef xlink:href="#t-play"/></roleSpec>
        <topicRef xlink:href="#t-hamlet" />
      </member>
      <member>
        <roleSpec><topicRef xlink:href="#t-character"/></roleSpec>
        <topicRef xlink:href="#t36" />
      </member>
    </association>

  </topicMap>
```

AsTMa=

Další jednoduchá textová syntaxe, jejímž autorem je Robert Barta.

HyTM, (HyTime TM)

Toto je úplně první syntaxe pro TM, byla založena na **HyTime** (Hypermedia/Time-based Structuring Language) značkovacího jazyku vycházející ho z *SGML*. V příštích verzích standardů TM už pravděpodobně nebude [12]. Autorem původního HyTM byl Martin Bryan [3].

CTM (Compact TM)

Nově připravovaná standardizovaná verze textové syntaxe vycházející z LTM, AsTMa= a N3[12].

3.3 Dotazovací jazyky

V každé mapě témat je spousty zajímavých informací, ale zpravidla jich je tolik, že pro uživatele nemají žádnou informační hodnotu, protože se v nich nemá šanci zorientovat. Proto na řadu přichází dotazovací jazyky, které nám umožní získat z mapy témat je ty témata či vztahy (informace), které nás zajímají. Tyto dotazovací jazyky můžeme chápat stejně jako dotazovací jazyky *SQL, ORACLE* pracující nad databázemi.

TMQL (TM Query Language)

TMQL je připravovaný dotazovací jazyk, který má být vydán jako standard pro dotazování TM ISO 18048. V současné chvíli je zde více návrhů jak by mělo TMQL ve finále vypadat. Původně měl být založen na některém s existujících jazyků a to *tologu,TMRQL a EmpolisTMQL*. [1] Nyní se bude na

implementací ISO TMQL angažovat společnost **Empolis**[10], ovšem nejsem si jist, jestli bude postaven přímo na eTMQL.

tolog

Tolog jeplně funkční dotazovací jazyk od firmy **Ontopia**, jeho funkčnost lze vyzkoušet v OKS Samplers (Bohužel v online demu možnost dotazování není). Sám jsem tolog používal ve své seminární práci ***4IZ440 – mapa témat a webová aplikace,*** **Knihovna médií**[28]a musím říci, že je to velmi jednoduchý a intuitivní dotazovací jazyk. Velkou výhodou tologu je, že dokáže dotazovat i RDF a klasické relační databáze[15]. Ukázka demonstruje dotaz na všechny opery, které byly inspirovány Shakespearem.

```
using m for i"http://psi.ontopia.net/music/"
using op for i"http://psi.ontopia.net/opera/"
using lit for i"http://psi.ontopia.net/literature/"
SELECT $OPERA,$WORK,$COMPOSER
FROM
 m:composed-by($OPERA : lit:work, $COMPOSER : m:composer),
 op:based-on($OPERA : op:result, $WORK : op:source),
 lit:written-by($WORK : lit:work, shakespeare : lit:writer)?
```

Zdroj: Ontopia omnigator – *query*, mapa témat *opera.ltm*

Empolis eTMQL

Jedná se o dotazovací jazyk implementovaný v softwaru **K42** firmou **Empolis**. Má stejný název jako připravovaný standard TMQL, ale nemá s ním nic společného (Zatím). Je navržen, aby se podobal jazykům SQL. Umí se dotazovat všechny aspekty TM. Vývoj byl zastaven a eTMQL bylo nahrazeno eRQL pro dotazování RDF.[10]

AsTMa?

Další experimentální dotazovací jazyk, jeho autorem je Robert Barta. AsTMa? umožňuje tvořit funkce. Více o tomto dotazovacím jazyce naleznete na stránkách autora http://astma.it.bond.edu.au/querying.xsp

XTMPath

Další experimentální dotazovací jazyk, jehož autorem je opět Robert Barta. Nejde ani tak o nový dotazovací jazyk jako spíše o využití dotazovacího jazyku XMLPath. Více informací o tomto jazyku najdete na http://topicmaps.bond.edu.au/docs/13?style=printable

A další

Toma, TMpath, TMRQL (od společnosti *NetworkPlanet*)

3.4 Klady, zápory a využití TM

3.4.1 Použití TM

Využití Topic Maps [12]

- Organizace rozsáhlých zdrojů informací (rejstříky v knihovnách)
- Shromažďování informací v distribuovaném prostředí
- Podpora navigace v aplikacích
- Vyhledávání v rozsáhlých databázích elektronických zdrojů
- TM se využívají v managementu znalostí pro práci se znalostmi

[28] Seminární práce bohužel není dostupná online – rok vytvoření 2007

- Kdekoliv, kde potřebuji zachytit vlastnosti o předmětu a vztahy mezi nimi. TM mohou dokonce v některých případech dobře nahradit relační databáze.
- integrace knihovních katalogů
- sémantické portály-kde *mapa témat slouží k definování informační architektury webu. Každé stránce odpovídá jedno téma. Navigace pro snadný přechod na související informace je na stránky automaticky vkládána na základě mapy témat*[12]
- E-Learning

Konkrétní aplikace pro práci s TM

Rozsáhlejší seznam nástrojů, aplikací a výrobců, kteří se zabývají TM, najdete na adrese http://www.topicmap.com/

- **Ontopia OKS**, kompletní rámec nástrojů pro práci s TM - komerční
- **Ontopia OKS Samplers**, nástroje pro editaci, prohlížení, dotazování a vizualizaci TM zdarma[12]
- **Empolis K42**, komerční
- **NetworkedPlanet – Tmcore05**, běží na .NET
- **TM4J –** Open-sourcový java TM engine, umí i některé nestandardizované technologie jako *tolog* a *LTM*
- **ZTM –** Nadstavba nad *Zope* pro tvorbu webu na základě map témat[12].

3.4.2 Klady

Jednou z největších výhod Topic Maps je, že pracuje na vyšší sémantické úrovni, tedy že přímo v mapě témat je uložená informace, že toto je název, tamto role ve vztahu a ono výskyt (na rozdíl od RDF kde je všechno objekt ve výroku). Dalším obrovským kladem je propracovaný systém identifikace témat tedy rozlišování předmětu, který téma reprezentuje na zdroj a předmět ze skutečného světa a jejich identifikaci na základě URI zdroje či URI jednoho nebo více lokátorů předmětu. Dále různé témata můžou mít více názvů či jiných charakteristik, které se dají filtrovat pomocí oborů pravdivosti, tedy systém zvolí nejvhodnější a zobrazí je. TM jsou tedy vhodné pro uchování znalostí v mnohojazyčné verzi. Většina výhod TM vychází z jejich složitějšího datového modelu a jejich samotné koncepce jako např. možnost N-árních vztahů, kde je orientace vztahu dána rolemi. Další obrovskou výhodou TM je naprosto jednoduchá reifikace, tedy vytvoření tvrzení o tvrzení (vytvoření tématu z čehokoli, co je v mapě témat). Další ohromně užitečnou vlastností je, že se dají mapy témat slučovat podle identity a tvoří agregovaný zdroj informací. Pro mě osobně je velkou výhodou pohodlnost s jakou lze s TM pracovat a jejich přehlednost při vizualizaci a procházení znalostí v těchto mapách.

3.4.3 Zápory

Přestože mají TM spoustu obrovských výhod, mají samozřejmě i nějaké nevýhody. Bohužel mě napadá mě jen špatná možnost odvozování a nevhodnost pro strojové zpracovávání informací v mapách, která je dána zejména určením TM pro lidi a její vyšší sémantická úroveň.

4 Rozdíly mezi RDF a TM

Nebylo jednoduší a efektivnější vytvořit pouze jeden standard namísto dvou podobných (RDF a TM). Odpověď není úplně jednoznačná. Na jednu stranu by to byl jistě přínos, ovšem vzhledem k tomu, že obě platformy vznikly v jiném čase a na půdě jiných institucí to ani nebylo možné. Jak říká sám pan Kosek, J „V okamžiku*, kdy se o sobě tyto komunity dozvěděly, bylo už pozdě na sjednocení přístupu*“[12]. Pravděpodobně je to takto lepší, neboť obě platformy mají přece jen trochu jiné

uplatnění. RDF vzniklo především pro strojové zpracování informací, zatímco TM jsou určeny k reprezentaci znalostí pro lidi. Proto mají tyto platformy u různé silné a slabé stránky. Např. RDF umožňuje odvozování a více propracované ontologie, což není v TM myslitelné. Mapy Témat zase umožňují efektivní slučování map na základě identifikátorů témat.

Než si vysvětlíme princip vzájemné transformace, bude nutné uvést rozdílné znaky technologií. Na některé rozdíly mezi RDF a TM jste po přečtení předešlých kapitol přišli určitě sami. Přestože jsou si obě technologie velmi podobné, jsou určeny k úplně jinému použití. Což je hlavním problémem, který znemožňuje tyto technologie sloučit. Topic maps, jak už bylo dříve řečeno, byly stvořeny jako vysoko-úrovňová podpora vyhledávání ve velkém množství informačních zdrojů (TM jsou generalizace rejstříku na konci knihy[12]). Zatímco RDF je určeno jako základní kámen sémantického webu, díky poskytování strukturovaných metadat o informačních zdrojích a díky vytvoření platformy pro logické odvozování. Celá tato kapitola je s výčtem rozdílů je převzata z článku [7] Larse Garshola.

4.1 Reprezentace věcí symboly

Jak RDF, tak TM jsou technologie založené na identitách. To znamená lidsky řečeno, že klíčovým konceptem je ***symbol*** reprezentující identifikovatelnou ***věc***, o které můžeme něco tvrdit. Ještě lidštěji řečeno vytvořím ***symbol***, který říká ***„toto reprezentuje osobu“***, poté můžu k tomuto symbolu přiřadit název ***Shakespeare*** a spojit tento jej s jiným symbolem ***knihy*** přes vztah ***„napsal“***.

- V TM je termín pro věc jednoduše **předmět (Subject)**, a symbolem představující předmět v mapách témat je **téma (Topic)**.
- V RDF je pojem pro věc **zdroj (Resource)** a termín, který reprezentuje zdroj v RDF je **uzel (Node)**.

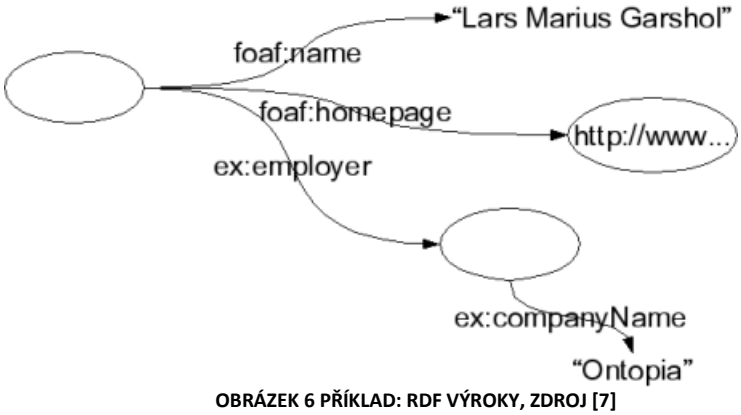
TABULKA 1 VZTAH POJMŮ V RDF A TM, ZDROJ [7]

Termín	Topic Maps	RDF	Soulad
Věc	Předmět (Subject)	Zdroj (Resource)	Přesný
Symbol	Téma (Topic)	Uzel (Node)	Těsný

4.2 Výroky, přiřazování

Zatím jsme si řekli o tom, jak tyto standardy používají symboly k reprezentaci věcí. To podstatné je však jakým způsobem RDF a Topic Maps mohou o těchto věcech něco říkat. Zde se ukazuje první rozdíl mezi rodinami standardů. RDF může tvořit výroky o věcech pouze jedním způsobem, kdežto TM mají tři různé způsoby (3 různé charakteristiky témat – názvy, výskyty a vztahy).

Jediný způsob přiřazení v RDF je **výrok (statement)**. Výrok je vztah **subjektu** (uzel, o kterém se vyjadřujeme) k **objektu** (uzel který je hodnotou) daný vlastností (predikátem). Objekt nemusí být pouze *zdroj*, ale může jím být i *literál* viz kapitola o RDF. Příklad RDF výroků o panu L.M.Garsholovi můžete vidět na obrázku níže.



OBRÁZEK 6 PŘÍKLAD: RDF VÝROKY, ZDROJ [7]

Jak bylo již řečeno, TM mají 3 různé způsoby jak vytvořit přiřazení a to jsou názvy, výskyty a role ve vztazích.

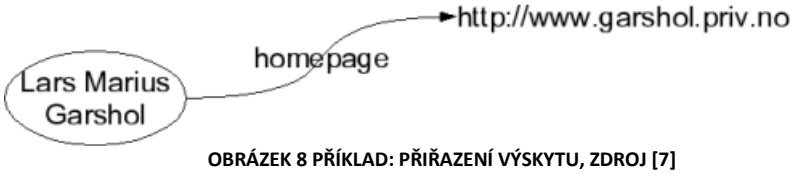
Názvy (Names)

Název je nejjednodušší přiřazení v TM. Název je textová hodnota (string), ale může být kvalifikována (může obsahovat obor platnosti). Toto přiřazení má v TM zvláštní postavení, jelikož reprezentuje téma navenek a je přímo v datovém modelu. V RDF se přiřazení názvu provádí použitím vlastnosti, která název vyjadřuje (které vlastnost to bude, se liší na použitém RDF slovníku[29]). Obrázek zobrazuje mapu témat s jedním tématem, které reprezentuje L.Garshola. toto téma má přiřazeno jeden název, který je vidět přímo na tématu.

OBRÁZEK 7 - PŘÍKLAD: TÉMA S NÁZVEM, ZDROJ [7]

Výskyty (Occurences)

Výskyty mohou být literály přímo v mapě témat, nebo URI odkazy na informační zdroje mimo mapu. Jak jsme si už řekli v kapitole 3, výskyty mají typ a tímto typem je téma ve stejné mapě témat. Na výskytech je zajímavé, jsou strukturálně stejné jako RDF vlastnosti (když pomineme podporu oborů platnosti). Díky této podobnosti (jak se později přesvědčíme) je možné konvertovat RDF a TM oběma směry. Každopádně i výskyty se drobně liší od RDF vlastností a to tím že RDF vlastnosti jsou sémanticky méně omezené. Příklad ukazuje přiřazení výskytu k tématu.

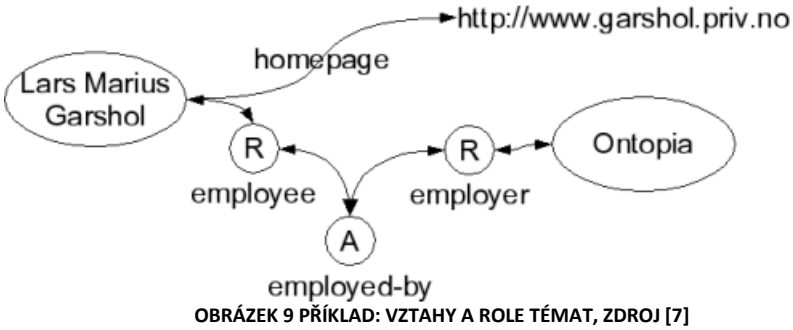


OBRÁZEK 8 PŘÍKLAD: PŘIŘAZENÍ VÝSKYTU, ZDROJ [7]

Vztahy (Associations)

Konečně můžeme také tvořit vztahy mezi tématy. Připomínám, že vztah má typ, kterým je téma ze stejné mapy. Každé **téma** ve vztahu se tohoto vztahu účastní jako hráč **role vztahu**. Těchto rolí ve vztahu může být neomezeně viz kapitola 3, na rozdíl od RDF, kde je to naprosto vyloučené. Příklad ukazuje vztah *employed_by* mezi osobou *L.Garshola* a jeho zaměstnavatelem spol. *Ontopií*. Může si všimnout rolí *emploeyee* a *employer*.

[29] V obrázku 5 to byla např. vlastnost *foaf:name*



OBRÁZEK 9 PŘÍKLAD: VZTAHY A ROLE TÉMAT, ZDROJ [7]

Na příkladu si můžete všimnout, jak se liší vztah mezi tématy od výskytu. Ale podstatnější je zde rozdíl jakým se tvoří přiřazení vztahu mezi tématy v TM a jak se tvoří výroky v RDF. V TM role reprezentují zapojení témat ve vztahu a platí **obousměrně** (tedy že Ontopia je *zaměstnavatel* Larse a zároveň, že Lars je zaměstnanec Ontopie (pozn. autora – je jejím spoluzakladatelem)). V RDF jsou vlastnosti **orientované** a platí pouze **jednosměrně**. Tedy v RDF nejde jedním výrokem říci, že je Ontopie zaměstnavatelem Larse a zároveň, že je u ní Lars zaměstnán, proto v RDF vznikají inverzní vztahy.

4.3 Identita

Identifikace je o zjišťování, jestli dva různé symboly nereprezentují stejnou věc. Jak RDF, tak i TM jsou navržené, aby identifikaci podporovaly, ale každá jí podporuje trochu jinak. V **RDF** existují 3 typy uzlů

- **Literály**
- **URI uzly** – tyto uzly jsou jasně identifikovatelné, pokud máme dva uzly se stejným URI, reprezentují oba jeden zdroj.
- **Prázdné uzly (blank nodes)** – Prázdné uzly jsou uzly, které nejsou označeny URI. Tento příklad je vidět na Obrázek 6. Jediný způsob, jak v RDF zjistit jakou věc prázdný uzel reprezentuje, je podívat se na výroky o tomto uzlu. Bohužel v RDF neexistuje žádná standardizovaná metoda ke zjištění, které prázdné uzly reprezentuj stejnou věc, ačkoli vysoko-úrovňové slovníky jako OWL to umožňují (viz kapitola 5.4).

V **TM** mohou být témata také jako prázdné uzly, ale v tématu mohou být přiřazeny URI pro **indikátory objektu (Subject identifier)**, který věc identifikují. Pokud nejsou žádné indikátory přiřazeny, je téma přesně jako prázdný uzel v RDF.

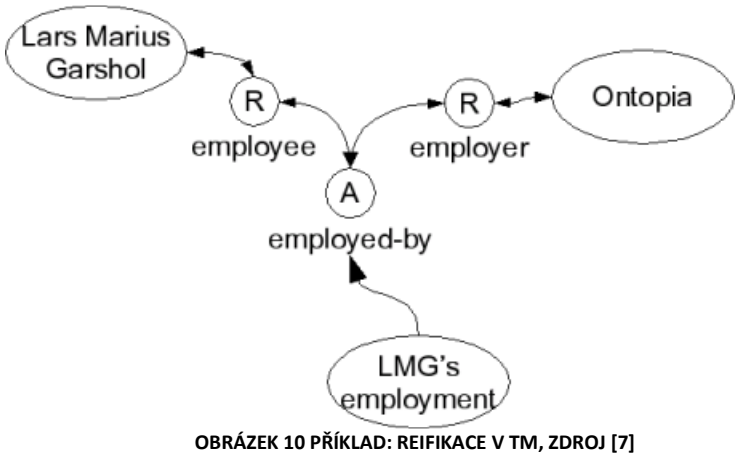
Jak bylo řečeno v kapitole 3, TM mají vlastní propracovaný systém identifikace předmětů. Rozlišují totiž předměty a informační zdroje. V RDF není na první pohled jasné, co URI reprezentuje. Například zda URI *„http://cs.wikipedia.org/wiki/William_Shakespeare“* reprezentuje stánku na wikipedii nebo samotnou osobu Williama Shakespeara. Samozřejmě se tento problém dá řešit na úrovni ontologie, když přiřadíme zdrojům třídy `Document` a `Person`. Toto je ožehavý problém interoperability[30] mezi oběma formáty.

4.4 Reifikace

O reifikaci už byla řeč v kapitole 3 o TM. Ve zkratce jde o vytvoření **tvrzení o tvrzení**. Jestli můžeme vytvořit symbol, který představuje přiřazení, můžeme tedy tento symbol použít. Vytvoření symbolu, který reprezentuje přiřazení je právě to, čemu se říká reifikace.

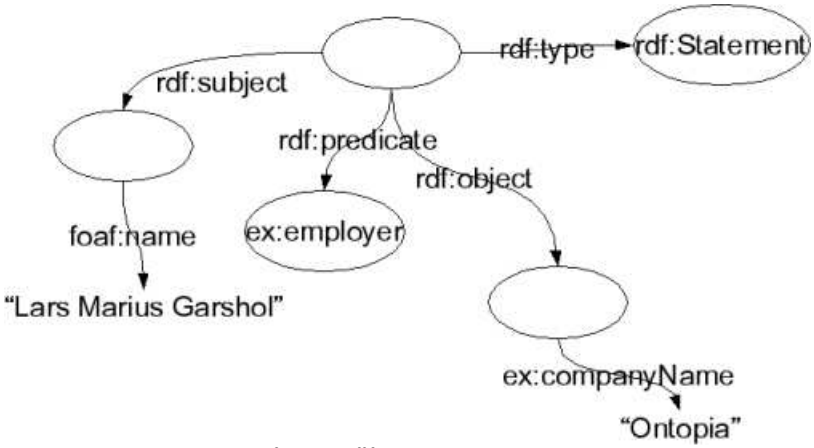
[30] Pro interoperabilitu je třeba sáhnout do vyšší jazykové úrovně a znát vhodné ontologie. [SVÁTEK, Vojtěch]

Jediné co v TM musíte k reifikaci udělat je, vytvořit téma a jako indikátor (subject identifier) použít název, výskyt nebo vztah který chceme reifikovat. Tuto situaci ukazuje níže uvedený příklad.



OBRÁZEK 10 PŘÍKLAD: REIFIKACE V TM, ZDROJ [7]

V RDF je reifikace mnohem složitější. Nejdříve je třeba vytvořit **prázdný uzel**, který je typu `rdf:Statement`. Poté se díky speciálnímu slovníku přidají uzly **subjekt, predikát a objekt**, které představují trojici výroku, který chceme reifikovat. Sice to není nijak elegantní řešení, nicméně je funkční. Tento nově vzniklý uzel má však jinou strukturu nežli původní uzel, což velmi komplikuje práci s reifikovanými výroky. Proto se v RDF reifikace používá jen velmi zřídka. Další nehezký jev je, že se na rozdíl od TM reifikací mění struktura RDF. Příklad ukazuje reifikaci výroku o Larsovo zaměstnání.



OBRÁZEK 11 PŘÍKLAD: REIFIKACE V RDF, ZDROJ [7]

4.5 Kvalifikace (Obory platnosti)

Občas je třeba vytvořit přiřazení, ve kterém je patrné, kdo tento výrok vyřkl, kdo tvrdí že je pravdivý, jaký je zdroj tohoto tvrzení či v jakém kontextu platí.

V mapách témat toto umožňuje vestavěná vlastnost TM nazvaná **scope (obor platnosti)**. O této vlastnosti jsme si povídali v kapitole 3.

V RDF neexistuje žádná podobná vlastnost, která by kvalifikaci umožňovala, tedy pokud pomineme, že literály mohou obsahovat jazykové identifikátory (což je jistý druh kvalifikace). Jediný způsob jak kvalifikovat výroky v RDF je použitím reifikace, což je jak jsme si ukázali výše v praxi dost neohrabané řešení.

Hlavní problémem je, že RDF **nemá identitu výroků**, což znamená, že nelze vytvořit zdroje, které by je reprezentovali[31]. Toto je jeden z **největších rozdílů mezi TM a RDF**.

Pro úplnost je třeba dodat, že se v současné době pracuje na rozšířeních RDF označovaných jako „named fraphs“, „annotated graphs“ či „networked graphs“, které tuto možnost poskytují [SVÁTEK, Vojtěch]. Více informací najdete na jednom z center pro tuto problematiku a to http://www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/Research/NetworkedGraphs.

4.6 Typy a podtřídy

Jedno z nejdůležitějších přiřazení, které se provádí je přiřazení **typu (type)**. Což nám v příkladu umožní říci, že Shakespeare je jiný druh věci než divadelní hra a že se na každý tento předmět vztahují jiná pravidla. Navíc bychom mohli chtít poznamenat, že divadelní hra je **podtřída (subclass)** díla. Tedy že všechny divadelní hry jsou současně hry, ale existují i jiné hry, které nejsou divadelní (např. opery).

- K určení typu se v RDF používá vlastnost `rdf:type`, která je součástí samotného RDF, a k určení podtřídy se používá vlastnost `rdfs:subClassOf`, která je spíše součástí RDF schema nežli RDF.
- V TM je situace podobná, k určení typu poskytují TM předdefinované vztahy `type-instance` a `supertype-suptype`, jediný rozdíl je, že jsou tyto vztahy součástí samotného standardu TM a že TM raději vlastnost nazývají `subtype` namísto `subClassOf`.

4.7 Slučování

Slučování je proces, který slučuje symboly, které reprezentují stejné věci. V RDF jsou to uzly, které mají stejný **literál**, nebo používají stejné **URI**. Jazyky pro ontologie jako OWL či DAML+OIL navíc poskytují mechanismus, který umožňuje identifikovat uzly podle jejich výroků. V Topic Maps se slučují jednoduše témata, které mají stejné **identifikátory předmětu (subject identifier)** či **lokátory předmětu (subject locator)**. Navíc stejně jako u RDF nové standardy TM umožňují identifikovat témata na základě přiřazení jejich charakteristik.

5 Princip vzájemné transformace mezi technologiemi

Nyní když jsme si vysvětlili oba modely, můžeme se konečně pustit do samotné transformace mezi formáty. Nejdříve si řekneme proč je vlastně potřeba takovéto konverze. Když už známe fundamentální rozdíly mezi oběma technologiemi, a můžeme se rovnou podívat na princip konverze. Princip, který zde budu prezentovat, vymyslel a publikoval **GARSHOL, Lars Marius** v článku [7], který budu v celé této kapitole hojně citovat (Dalo by se říci, že se jedná o volný překlad). Jedná se o jediný článek takového rozsahu, který se zabývá transformací mezi těmito formáty.

5.1 Proč transformovat

Dokaď nelze oba formáty sloučit, měli bychom být aspoň schopni přesouvat data z jednoho formátu do druhého. Je nutné, aby konvertované data byly úplné a obsahovali stejné znalosti a informace, které byly přístupné v původním modelu. Nevětší důvod proč vlastně přesouvat data mezi formáty je interoperabilita mezi aplikacemi a výměna informací mezi subjekty.

[31] Aniž by změnili strukturu těchto tvrzení.

5.2 Transformace dat z RDF do TM

K transformaci mezi formáty je nutné vytvořit takzvané **mapování (Mapping)** z jednoho formátu do druhého. V tomto případě z RDF do TM. Jak se ukázalo, vytvoření takového mapování je složitější než se může na první pohled zdát. Ovšem když se budeme držet daného postupu, ukáže se nám postupně celé. Tento postup vypadá následovně:

1. Modelování TM v RDF
2. Tvorba mapování pro specifické slovníky (schémata)

5.2.1 Modelování TM v RDF

Modelování TM v RDF znamená používat **RDF uzly** tak, aby představovali **názvy, výskyty a vztahy TM** a **RDF tvrzení** k přiřazení **typů a oborů pravdivost** k uzlům. K tomuto účelu vytvořil Lars M. Garshol vhodné RDF schema, které publikuje v [8], pro snazší pochopitelnost doporučuji pročíst. Toto RDF schema popisuje RDF slovník pro popis mapování RDF schémat na Topic Maps jako sadu PSI. Díky používání tohoto mapování lze RDF data konvertovat do TM a RDF slovníky mohou definovat své vlastní mapování na TM. To umožňuje, aby byly stejné slovníky použity v RDF i v TM zároveň. Tento RDF slovník je vyvíjen jako část práce[7] a je součástí OKS a Omnigatoru.

Bohužel neexistuje jiný způsob jak přirozeně použít při modelování TM v RDF reifikaci a obory pravdivosti, než použití RDF reifikace, která není uspokojujícím řešením.

5.2.2 Tvorba mapování pro specifické slovníky

Jako největší problém se zde při konverzi ukazuje, že nemáme všechny potřebné informace k vytvoření obecného mapování. Podívejme se na níže uvedenou RDF trojici (subj, pred, obj).

```
(http://example.com/X,     http://example.com/Y,        "text")
```

Jedná se o výrok, který přiřazuje subjektu **X** vlastnost **Y** s **textovou hodnotou** (literálem), ale už není vůbec patrné, co toto přiřazení znamená[32]. Jedná se snad o název předmětu X, či pouze o nějakou jeho vlastnost? Bez této informace nemůžeme vytvořit mapování TM. Podívejme se na další výrok.

```
(http://example.com/X, http://example.com/W,  http://example.com/Z)
```

Přestože je tento výrok v RDF naprosto jednoznačný, v TM není vůbec jasné, zdali se jedná o přiřazení výskytu či o vztah mezi tématy[32]. A pokud se jedná o vztah, kde jsou role objektu a subjektu?

Přestože se nám nepovedl vytvořit obecné mapování, máme klíč k vytvoření specifického mapování. Proč tedy neposkytnout konvertoru nezbytné rozlišení přiřazení? Ovšem další problém je s mapováním identifikačního **URI**, pokud je pouze jedno. Jistě bude to určitě **identifikátor objektu** nebo **lokátor objektu**, ale který že? Tato informace musí být také poskytnuta mapováním.

Když už jsme se jednou rozhodli co dělat se subjektem, ukazuje se, že ve většině případů k dokončení mapování potřebujeme víc informací, které by nám měli říci, jak mohou vypadat uzly objektů. Tabulka stručně ukazuje, jaké typy uzlů se mohou vytvořit pro tyto objekty.

[32] V obou příkladech se ve skutečnosti dá poznat, co výroky vyjadřují ze sémantiky jejich predikátů. Je ovšem nutné ovládání ontologických jazyků a použití vhodných ontologií, což je jistě mnohem složitější, nežli využívání pevných prvků datového modelu RDF a TM. [SVÁTEK, Vojtěch]

TABULKA 2 SHRNUTI POTREBNYCH INFORMACI PRO MAPOVANI A MOZNE TYPY UZLU, ZDROJ [7]

Mapování pro	Informace potřebná k mapování	Možné typy uzlů
Název (Name)	Obor pravdivosti (Scope)	Literál
Výskyt (Occurence)	Typ, obor pravdivosti	Literál nebo URI
Association (Vztah)	Typ, Obor pravdivosti, typy rolí	URI nebo prázdný
Lokátor předmětu (Subject locator)		URI
Identifikátor předmětu (subject identifier)		URI

Teď už jsme jen krůček od toho, na co chceme výrokem mapovat. To záleží především na použitém predikátu, který nám říká jak výrok interpretovat. To znamená, že můžeme použít RDF k anotaci (označení) vlastnosti použité ve slovníku (schématu) s informacemi potřebnými pro použití v TM. K tomuto účelu byl vytvořen **mapovací slovník (Mapping vocabulary)**, který obsahuje RDF vlastnosti, které jsou uvedeny níže.

- `rtm:maps-to` se používá k určení na co se RDF vlastnost mapuje. Možné hodnoty jsou `rtm:basename, rtm:occurrence, rtm:association, rtm:source-locator, rtm:subject-identifier, rtm:subject-address, a rtm:instance-of`. Poslední je zkratkou pro tvorbu vztahu type-instance
- `rtm:type` se používá pro určení typu výskytu, ke kterému se výrok mapuje.
- `rtm:in-scope` se používá pro specifikaci oboru pravdivosti názvu tématu, výskytu či vztahu ke kterému se výrok mapuje.
- `rtm:subject-role` a `rtm:object-role` se používají pro určení typů rolí, které subjekty a objekty hrají v tvrzení, které se mapuje na vztah.

RDF schema s tímto mapováním vytvořil Lars Marius Garshol a najdete jej na [8], včetně detailnějšího popisu schéma (viz předchozí kapitola 5.2.1 Modelování TM v RDF).

Největší výhodou tohoto přístupu je nejspíš, že umožňuje RDF slovníkům být zároveň slovníky TM. Když publikujeme RDF Schema pro slovník, mapování na TM může být publikováno ve stejnou chvíli a může být součástí stejného balíčku bez potřeby speciální syntaxe potřebné k mapování (RDF syntaxe k tomu bohatě stačí). A především jsou použity stejné URI jak v TM, tak v RDF, které ji reprezentuje.

V příkladu jsou uvedeny informace o Larsovi v FOAF slovníku (o autorovi tohoto principu, zdroj [7])

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

<foaf:Person rdf:ID="larsga">
  <foaf:name>Lars Marius Garshol</foaf:name>
  <foaf:nick>larsbot</foaf:nick>

  <foaf:mbox rdf:resource="mailto:larsga@garshol.priv.no"/>
  <foaf:homepage rdf:resource="http://www.garshol.priv.no/" />

  <foaf:knows>
    <foaf:Person>
      <foaf:mbox rdf:resource="mailto:grove@ontopia.net"/>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
</rdf:RDF>
```

Tato ukázka je z větší části sebe vysvětlující, s výjimkou vlastnosti `foaf:nick`, která obsahuje Larsovu přezdívku. Mapování (nejen) pro tuto ukázku s částí slovníku FOAF můžete najít níže, mapování používá výše uvedené schéma. (Dejte si pozor, předpony jmenných prostorů (namespace prefixes) byly použité v `rdf:about` a `rdf:resource` atributech pro lepší čitelnost, ale jsou syntakticky špatně.)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:rtm="http://psi.ontopia.net/rtm/#">

<rdf:Description rdf:about="rdf:type">
  <rtm:maps-to rdf:resource="rtm:instance-of"/>
</rdf:Description>

<rdf:Description rdf:about="foaf:name">
  <rtm:maps-to rdf:resource="rtm:basename"/>
</rdf:Description>

<rdf:Description rdf:about="foaf:nick">
  <rtm:maps-to rdf:resource="rtm:basename"/>
  <rtm:in-scope rdf:resource="foaf:nick"/>
</rdf:Description>

<rdf:Description rdf:about="foaf:mbox">
  <rtm:maps-to rdf:resource="rtm:occurrence"/>
</rdf:Description>

<rdf:Description rdf:about="foaf:homepage">
  <rtm:maps-to rdf:resource="rtm:occurrence"/>
</rdf:Description>

<rdf:Description rdf:about="foaf:knows">
  <rtm:maps-to rdf:resource="rtm:association"/>
  <rtm:subject-role rdf:resource="foaf:Person"/>[33]
  <rtm:object-role rdf:resource="foaf:Person"/>[33]
</rdf:Description>
</rdf:RDF>
```

Toto je vše co potřebuji pro transformaci tohoto RDF do TM. Pod textem je uvedená ukázka transformovaného RDF v syntaxi XTM. (zkratky jemných prostorů byly opět použity pro větší čitelnost)

[33] Tyto informace jsou dostupné v RDF slovníku pro FOAF z vlastností `rdfs:domain` a `rdfs:range`

```
<topic id="id35">
    <instanceOf>
      <subjectIndicatorRef xlink:href="foaf:Person"/>
    </instanceOf>
    <baseName>
      <baseNameString>Lars Marius Garshol</baseNameString>
    </baseName>
    <baseName>
      <scope>
        <subjectIndicatorRef xlink:href="foaf:nick"/>
      </scope>
      <baseNameString>larsbot</baseNameString>
    </baseName>
    <occurrence>
      <instanceOf>
        <subjectIndicatorRef xlink:href="foaf:email"/>
      </instanceOf>
      <resourceRef xlink:href="mailto:larsga@ontopia.net"/>
    </occurrence>
    <occurrence>
      <instanceOf>
        <subjectIndicatorRef xlink:href="foaf:homepage"/>
      </instanceOf>
      <resourceRef xlink:href="http://www.garshol.priv.no"/>
    </occurrence>
  </topic>

  <association>
    <instanceOf>
      <subjectIndicatorRef xlink:href="foaf:knows"/>
    </instanceOf>
    <member>
      <roleSpec><subjectIndicatorRef
         xlink:href="foaf:Person"/></roleSpec>
      <topicRef xlink:href="#id35"/>
    </member>
    <member>
      <roleSpec><subjectIndicatorRef
         xlink:href="foaf:Person"/></roleSpec>
      <topicRef xlink:href="#id52"/>
    </member>
  </association>
```

Jak je vidět, konverze splňuje kriterium přirozenosti. Lars použil tuto metodu úspěšně na více RDF dat a podle [7] je přesvědčen, že **tato metoda stačí** pro většinu **úloh konverze RDF do TM**.

5.3 Transformace dat z TM do RDF

Zatím jsme si ukázali jak převést data z RDF do TM, aby bylo řešení kompletní, je třeba konvertovat data i opačně. Lars Garshol preferuje TM a raději bude spravovat své FOAF data v TM a později je konvertuje do RDF pro sdílení s RDF uživateli. Na příkladu níže jsou stejné FOAF data jako v předchozím příkladě, ovšem v TM notaci LTM, pro větší názornost.

```
[larsga : person = "Lars Marius Garshol"
    = "larsbot" / nick]
{larsga, mbox, "mailto:larsga@ontopia.net"}
knows(larsga : person, grove : person)

{grove, mbox, "mailto:grove@ontopia.net"}
```

Když jsme převáděli RDF do TM, problémem byly pouze chybějící informace nezbytné k mapování. Může se zdát, že by to opačným směrem mělo jít bez problémů, protože nezbytné informace už máme. Ovšem v tomto případě (výše zmíněném) nám pro změnu chybí správné URI FOAF slovníku, které vypadají takto:

```
[person = "Person" @"http://xmlns.com/foaf/0.1/Person"]
[nick = "Nick"     @"http://xmlns.com/foaf/0.1/nick"]
[mbox = "Email"    @"http://xmlns.com/foaf/0.1/mbox"]
[knows = "Knows"   @"http://xmlns.com/foaf/0.1/knows"]
```

Teď máme všechny informace, které potřebujeme k transformaci tohoto TM do FOAF RDF. Začneme s tématem `larsga`, které se změní v prázdný uzel. Typ přiřadíme díky vlastnosti `rdf:type` (tato vlastnost je součást RDF). Ovšem je tu háček, objektem tohoto výroku o typu je téma person. V tomto případě použijeme k jeho reprezentaci URI `http://xmlns.com/foaf/0.1/Person` z tabulky. Může ovšem dojít k situaci, že je téma součástí mapy, jaké URI se pak použije? Je to prosté – použije se buď jeho identifikátor, nebo jeden z jeho lokátorů (libovolný, ostatní budou označeny jako `owl:equivalentClass`).

Dalším krokem bude přiřazení jména. Zde je menší problém. Nevíme jakou vlastnost (predikát) máme použít pro jméno, protože RDF zdroje různých typů mají většinou jiné vlastnosti pro názvy. Zatím to přeskočíme a budeme si pamatovat, že je opět třeba vytvořit specifické mapování. Tento problém se bude pravděpodobně časem stávat méně zásadním, protože se stále častěji používají jako predikát názvu vlastnosti z oblíbených ontologií FOAF a Dublin Core.

Jméno `larsbot` je v mapě témat název s oborem pravdivosti nick. Jelikož nick není tématem v stejné mapě, k přiřazení názvu nick k Larsovi použijeme vlastnost `http://xmlns.com/foaf/0.1/nick`, stejně jako jsme přiřadily typ.

Nyní se podíváme na výskyt `mbox`. Přiřadíme jej stejným způsobem, objektem zde bude literál s emailovou adresou. Stejným způsobem se konvertují všechny výskyty, s jediným rozdílem, že vytvoří jako objekt literál nebo URI uzel, což závisí na tom, jestli je výskyt vnitřní nebo vnější.

Teď už pouze stačí převést vztah `knows` na výrok, k vytvoření vlastnosti se použije stejný mechanismus jako v předchozích případech. Zde je opět menší problém. V TM má vztah knows obě role typu person, čili nejde určit, co bude RDF subjekt a co objekt. Problém se vyřeší tak, že se subjekt vybere libovolně[34], a vlastnost se označí jako `owl:SymmertricProperty`. Pokud budu konvertovat obecný nesymetrický vztah TM, bude situace obtížnější. V TM je orientace vztahu dána rolemi, ovšem u RDF velmi záleží, co bude subjektem a co objektem, protože vztah nebude platit opačným směrem (Např. výrok „<Kouba> <rad_cte> <Shakespeare>“ nejde chápat i v opačném směru).

[34] Ideálnější by samozřejmě bylo vytvořit pravidlo, že první role bude vždy subjektem a druhá objektem, ovšem v tomto případě je to jedno, jelikož je vlastnost `knows` symetrická. Má tedy v obou směrech stejný význam.

Z tohoto příkladu je vidět, že není možné vytvořit obecné mapování TM na RDF, ale je možné vytvořit specifické mapování podobně jako u mapování z RDF na TM. Bylo by ideální, kdyby šlo mapování z RDF na TM použít i obráceně pro TM.

Jistě v praxi je ještě spousty případů, které musí být podobně ošetřeny, ale jak vidíte, je možné vytvořit specifického slovníky pro mapování, které nám **umožní TM do RDF transformovat**[7]. Ještě je třeba hodně zapracovat na vývoji specifického mapování, ale některé problémy jsou už vyřešené:

- Vícenásobné URI pro jedno téma lze vyřešit použitím RDF vlastností ekvivalence, která je v OWLu.
- Vztahy s více rolemi, jdou převést na jednotlivé zdroje, jejíž typ je *vztah*, a každá role tvoří vlastní výrok s typem *role* a vztahem jako subjektem.
- Reifikace a obory pravdivosti mohou být v RDF reprezentované reifikací. Pro reprezentaci oborů pravdivosti je třeba definovat speciální vlastnost.
- Binární nesymetrické vztahy mohou být převedeny, pokud bude mapování obsahovat výchozí roli pro subjekt.
- převedení názvu může být uskutečněno, pouze pokud bude v mapování uvedena RDF vlastnost pro název zdroje.

Samozřejmě že zde nešlo postihnout všechny případy a speciality TM, jako např. unární vztahy a rozdílné názvy témat. Je třeba přijít s nějakými inteligentními pravidly, které by určily, jaké vztahy budou při konverzi vynechány, je pravděpodobné, že ani nebude potřeba všechny informace z TM do RDF.

5.4 Transformace OWL

Předchozí kapitoly popisovali převod dat mezi oběma formáty, ovšem k úplné interoperabilitě je třeba dokázat převést i omezení. Tyto omezení jsme si probrali již v kapitole 2.2.1. Pokud se podaří převést z RDF do TM i omezení, bude možno s výhodou používat validačních nástrojů pro ověření správnosti datových transformací. Tato kapitola je opět převzata z [7] L. Garshola.

Konverze OWL do TM a OSL (Ontopia Schema Language), je samozřejmě složitější než konverze RDF schémat, kterou popisuje L.Garshole v [7], jelikož OWL jde mnohem dál nežli RDFS a samozřejmě i dál než OSL. Tuto situaci by ještě mohl hodně změnit právě vznikající OWL 1.1 http://www.webont.org/owl/1.1/. Nejdříve si ukážeme možnosti webové ontologie OWL a jak by mohli být nejlépe reprezentovány v TM. Nyní si uvedeme hrubé shrnutí schopností jazyku OWL a jejich mapování pro TM a OSL.

- **Metadata** (data o datech)- OWL má schopnost poskytovat informace o ontologii, jako např. informace o verzi, vztahy s jinými ontologiemi a dokonce umožňuje importovat jiné ontologie. Tyto informace mohou být na TM mapovány přímo, jako jednoduché popisující vlastnosti.
- **Definice tříd** - `owl:Class` je podtřídou `rdf:Class`, navíc OWL dokáže s RDFS definovat třídy výčtem stejně tak jako množinou operací na existujících třídách. OSL nemá žádnou podobnou součást, a musí být o tuto schopnost doplněna. Toto je schopnost požadovaná od připravovaného TMCL, ale zatím ještě není přítomna.
- **Omezení hodnot** – Omezuje hodnoty vlastností z RDF schéma. Tyto hodnoty můžou být navíc omezeny v závislosti na typu třídy jejich subjektu. Příklad: OWL může říci, že hodnota (objekt) vlastnosti `hasMaker` subjektu `wine` musí být instance třídy `winery`. Je třeba rozlišovat globální a lokální omezování hodnot. Zatímco globální

omezování hodnot umí i samotné RDFS (pomocí vlastnosti `rdfs:range`), lokální [35] omezení je vymožeností jazyku OWL. Této vlastnosti neodpovídá žádná vlastnost v TM, protože TM mají různé charakteristiky, na rozdíl od RDF, kde jsou pouze výroky o zdrojích, je tedy nutno rozlišovat jaké hodnoty chceme omezovat.

- **Omezení početnosti** (Cardinality restrictions) – umožňuje omezit počet výroků určité třídy subjektu. Funguje stejně jako omezení v OSL, takže bude velmi snadné je mapovat.
- **Globální početnosti** – `owl:FunctionalProperty` je podtřídou `rdf:Property`, což je třída vlastností, která může mít pouze jednu hodnotu (Funkční závislost). Funguje stejně, jako když je nastavená početnost vlastnosti na 1. `owl:InverseFunctionalProperty` je jiná třída vlastnosti, kde dva různé subjekty nesmí mít za vlastnost stejnou hodnotu. Toto je ekvivalent **unikátní charakteristiky** (Unique charakteristic) v modelu **SAM**[36], takže nejlepší a nejjednodušší bude, když se třída této vlastnosti změní na *unikátní charakteristiku* modelu SAM.
- **Vztahy podtříd** – OWL používá vlastnost `rdfs:SubClassOf` z RDFS se stejnou sémantikou. Takže se bude mapovat stejně jako v předchozích případech.
- **Ekvivalentní třídy** – `owl:equivalentClass` je vlastnost, která nám říká, že dvě různé třídy jsou ekvivalentní (shodné), přestože můžou mít jiný význam. Např.: třídy *král* a *mužNosícíKorunu* jsou ekvivalentní třídy. Bohužel OSL nemá žádnou takovou vlastnost a mapuje se tedy jako vztah mezi dvěma tématy.
- **Disjunktní třídy -** Vlastnost `owl:disjointWith` vyjadřuje, že instance jedné třídy nemůže být nikdy instancí druhé. OSL tuto vlastnost zahrnuje také, ovšem funguje tak, že někdo musí dovolit, aby třídy mohli sdílet instance, což je standardně zakázané). Tato vlastnost se také mapuje jako vztah mezi tématy.
- **Objekt versus objektová vlastnost** – OWL definuje podtřídy `rdf:Property` (`owl:DatatypeProperty` a `owl:objectProperty`). První umožňuje jako hodnoty vlastností pouze literály a druhé dovoluje naopak pouze instance tříd (URI). Toto je hlavní podmnožina schopností poskytovaných vlastností `rdfs:range`.
- **Ekvivalentní vlastnost** – Vlastnost `owl:equivalentProperty` funguje stejně ekvivalentní třídy, ale vztahuje se na vlastnosti. TM nemají ekvivalentní vlastnosti, a proto je lze mapovat pouze jako vztahy se stejnou sémantikou.
- **Inverzní vlastnosti** – Vlastnost `rdf:inverseOf` vyjadřuje, že objekt vlastnosti je opakem jiného objektu. Tato vlastnost se dá v TM mapovat jako vztah mezi tématy, a není tedy potřeba.
- **Tranzitivní (transitive) a symetrické vlastnosti** - OWL definuje dvě třídy vlastností (`owl:TransitiveProperty` a `owl:SymmetricProperty`), které jsou obě podtřídou `owl:objectProperty`. TM neobsahují žádný ekvivalent těchto tříd, ale tyto třídy mohou být použity přímo pro určení typu vztahů mezi tématy.

Nyní si ukážeme příklad mapování pro část slovníku OWL, která může být vyjádřená v mapě témat, příklad je převzat z[7]. (Entity **&owl;** a další byly použity místo URI předpon, aby byl příklad srozumitelnější)

[35] Lokální znamená definované pro určitou třídu z domain.
[36] Více v seznamu použitých zkratek

```
<rdf:Description rdf:about="&owl;equivalentClass">
    <rtm:maps-to rdf:resource="&rtm;association"/>
    <rtm:subject-role  rdf:resource="&owl;Class"/>
    <rtm:object-role  rdf:resource="&owl;Class"/>
  </rdf:Description>

  <rdf:Description rdf:about="&owl;disjointWith">
    <rtm:maps-to rdf:resource="&rtm;association"/>
    <rtm:subject-role  rdf:resource="&owl;Class"/>
    <rtm:object-role  rdf:resource="&owl;Class"/>
  </rdf:Description>

  <rdf:Description rdf:about="&owl;equivalentProperty">
    <rtm:maps-to rdf:resource="&rtm;association"/>
    <rtm:subject-role  rdf:resource="&rdf;Property"/>
    <rtm:object-role  rdf:resource="&rdf;Property"/>
  </rdf:Description>

  <!—Zde se ošetřují třídy vlastností -->
  <rdf:Description rdf:about="&rdf;type">
    <rtm:maps-to rdf:resource="&rtm;instance-of"/>
 </rdf:Description>
```

OWL zde hraje dvě role, zaprvé poskytuje možnost více omezení a umožňuje nám vytvořit užší OSL schéma z RDF a zadruhé poskytuje sémantickou anotaci typů ve schéma.

5.4.1 Shrnutí

Lze s jistotou říci, že konverze omezení RDFS, OWL, OSL a budoucího TMCL nebude složitá. Naopak bude jednoduší než převod dat, protože slovníky těchto omezovacích jazyků jsou pevně dané a dá se k nim snadno vytvořit obecné mapování, které bude znovupoužitelné. Navíc se ukázalo, že schopnosti OWL, které jdou dál, než za hranice obyčejného omezování ve většině případů půjdou v TM přímo použít. Z tohoto důvodu to vypadá jako nejlepší řešení aby připravované TMCL podporovalo pouze omezení a nechalo ontologii pouze na OWL, který je, jak se ukázalo, v TM použitelný. [7]

5.5 Transformace RDF Schema

RDF schéma jde konvertovat mnohem snáz nežli OWL, neboť není zdaleka tak propracovaný a vyjadřující. Princip transformace je publikován v [7] Larse Garshola, zde ho uvádět nebudu, jelikož je to nad rámec této práce.

6 Existující nástroje pro konverzi

Nyní už jsme se už přesvědčili, že je konverze možná a to v obou směrech. Seznámili s principem této transformace a nyní se můžeme podívat na nástroje, které transformaci umožňují, a pokusíme se zhodnotit kvalitu této transformace. K tomuto účelu si vytvoříme jednoduchou XTM mapu témat bez použití hi-end features jako jsou reifikace a kvalifikace a ternární vztahy. Jako základ použiji část mapy témat obsahující informace o albech a skladbách z mé loňské seminární práce[37].

[37] o mé seminární práci **media library** byla řeč v kapitole 3.3

Zatím se mi podařilo najít pouze dva nástroje, které konverzi umožňují. První je **OKS** (viz kapitola 3.4.1), druhým je freewarový nástroj **TM4L (TopicMaps for E-Learning)**. Není vyloučeno, že nástrojů je více, ale sám Lars mi v emailu potvrdil, že o jiných nástrojích také neví. Zajímavost, která by Vám neměla uniknout, je **java třída RDFimporter**, jejímž účelem je konverze RDF dat do TM. Zdrojové kódy této třídy naleznete na http://tmapi-utils.cvs.sourceforge.net/tmapi-utils/tmapi-utils/src/org/tmapiutils/impexp/rdf/RDFImporter.java?revision=1.1&view=markup. Bohužel komerční TM nástroje od Mondeca a Empolis se mi nepodařilo nikde získat.

6.1 Nástroje pro konverzi TM do RDF

Nejprve se podíváme na transformaci TM do RDF, která má podle mého názoru větší významnost a to především kvůli vlastnímu určení TM. Tedy TM jsou určeny pro lidi, ale jejich informace jsou srozumitelné i pro stroje, ale většina RDF dat (určená pro stroje) pro člověka být srozumitelná nemusí (I když v TM budou).

6.1.1 OKS (Ontopia knowledge Suite) – Ontopoly

Prvním nástrojem, který umožňuje transformaci tímto směrem je komerční **OKS Ontopoly**. Tento nástroj je přehledný a velmi intuitivní, samotná konverze je dostupná v záložce export, stačí si vybrat t položek LTM,XTM,**RDF** a navíc si můžete vybrat, zda chcete exportovat i schéma mapy témat. Jedinou a to poměrně velkou nevýhodou je, že se při každém otevření a mapy v Ontopoly automaticky generují nové vnitřní identifikátory prvků mapy, a to navíc ještě náhodně. Aby byl příklad srozumitelnější snažil jsem v demonstrační mapě TMtest.xtm přejmenovat identifikátory ve smyslu `id112` na `skladba`, ovšem po opětovném načtení mapy se záměrem exportu do RDF, se klíče vygenerovali znovu a namísto očekávaného id `sklaba`, nebo méně očekávaného `id112` se objevilo naprosto neočekávané a nepochopitelné `id63`. S tímto problémem jsem se setkal už loni při zpracovávání své seminární práce, a v kapitole *známé chyby* jsem o něm zmiňoval, nyní už víme, čím tato chyba byla způsobená. Z tohoto důvodů jsem jak zdrojový XTM, tak i výsledný RDF soubor upravil až po konverzi. Další chybou, kterou jsem u OKS zaregistroval je špatná interpretace typu identifikátoru v notaci LTM. Přestože jsem v mapě témat nastavil u alb a skladeb identifikaci pomocí lokátoru předmětu, v LTM výstupu jsou označeny jako subject identifier tedy uvozeny znakem „%“ namísto „@“ pro subject locator.

Nyní se podívejme na demonstrační mapu bez schématu, vytvořenou v OKS v syntaxi LTM.

```
@"utf-8"
#VERSION "1.3"
/*  Generator: Ontopia Knowledge Suite (OKS)
    Date:      2007-12-27 13:17
*/
/* ----------------- TOPIC MAP ----------------- */
#TOPICMAP ~ TMTest
[TMTest = "TmTest"]
/* ----------------- ONTOLOGY ------------------ */
/* ----------------- Topic Types --------------- */
[album = "album"
    @"http://psi.example.org/album"]
[skladba = "skladba"
    @"http://psi.example.org/audiotrack"]
/* ----------------- Association Types --------- */
[ma-skladbu = "ma_skladbu"
    = "ma_skladbu" / album
    = "je_v_albu" / skladba
    @"http://psi.example.org/has_track"]
/* ----------------- Occurrence Types ---------- */
[autor = "autor"
    @"http://psi.example.org/autor"]
/* ----------------- INSTANCES ----------------- */
/* ----------------- Topics -------------------- */
/* -- TopicType: album -- */
[division-bell : album = "The Divison Bell"
    %"http://en.wikipedia.org/wiki/The_Division_Bell"]
   {division-bell, autor, [[Pink Floyd]]}

/* -- TopicType: skladba -- */
[cluster-one : skladba = "Cluster One"
    %"http://en.wikipedia.org/wiki/Cluster_One"]
[high-hopes : skladba = "High Hopes"
    %"http://en.wikipedia.org/wiki/High_Hopes_(Pink_Floyd_song)"]

/* ----------------- Associations -------------- */
/* -- AssocciationType: ma-skladbu */
ma-skladbu( division-bell : album, cluster-one : skladba )
ma-skladbu( division-bell : album, high-hopes : skladba )
```

Jedná se o jednoduchou mapu témat s jedním **albem**(The Division Bell), které má vztah ke 2 **písním**(High Hopes,Cluster One) a jeden **výskyt**(Pink Floyd). Nyní se můžeme podívat, jak dopadla transformace do RDF. Pro větší přehlednost jsem v zdrojovém kódu RDF nahradil URI *http://en.wikipedia.org/wiki/* za *@wiki,* přestože je to syntakticky špatně.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:j.0="http://psi.example.org/"
    xmlns:j.1="http://psi.ontopia.net/tminrdf/#" >
  <rdf:Description rdf:about="http://psi.example.org/autor">
    <rdfs:label>autor</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="http://psi.example.org/album">
    <rdfs:label>album</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="@wiki#High_Hopes_(Pink_Floyd_song)">
    <j.0:has_track rdf:resource="@wiki#The_Division_Bell"/>
    <rdfs:label>High Hopes</rdfs:label>
    <rdf:type rdf:resource="http://psi.example.org/audiotrack"/>
  </rdf:Description>

  <rdf:Description rdf:nodeID="A2">
    <rdf:predicate rdf:resource="http://www.w3.org/2000/01/rdf-
      chema#label"/>
    <rdf:subject rdf:resource="http://psi.example.org/has_track"/>
    <j.1:inscope rdf:resource="http://psi.example.org/audiotrack"/>
    <rdf:object>je_v_albu</rdf:object>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
     ns#Statement"/>
  </rdf:Description>

  <rdf:Description rdf:about="@wiki#Cluster_One">
    <rdf:type rdf:resource="http://psi.example.org/audiotrack"/>
    <j.0:has_track rdf:resource="@wiki#The_Division_Bell"/>
    <rdfs:label>Cluster One</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="http://psi.example.org/audiotrack">
    <rdfs:label>skladba</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="file:/C:/Program Files/oks-
     samplers/apache-tomcat/webapps/omnigator/WEB-
     INF/topicmaps/tmtest.xtm#TMTtest">
    <rdfs:label>TmTest</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="http://psi.example.org/has_track">
    <rdfs:label>ma_skladbu</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about="@wiki#The_Division_Bell">
    <j.0:autor>Pink Floyd</j.0:autor>
    <rdf:type rdf:resource="http://psi.example.org/album"/>
    <rdfs:label>The Divison Bell</rdfs:label>
  </rdf:Description>
</rdf:RDF>
```

Jak je vidět konverze je velmi zdařilá, v modelu jsou všechny informace, a struktura vypadá velmi podobně jako v případě syntaxe XTM. Za zmínku stojí především elegantní konverze vztahu `ma_skladbu,` který byl převeden na prázdný uzel a pomocí reifikace jí byl přidán dokonce i původní obor pravdivosti (scope), který dává tomuto vztahu název `je_v_albu` v kontextu skladby. Dokonce je definován jmenný prostor (namespace) pro PSI, která jsem svým tématům a vztahům přiřadil. OKS bohužel nepodporuje konverzi opačným směrem (aspoň ne v zkušební verzi OKS samplers), takže nebylo možno vyzkoušet kvalitu zpětným převedením do TM.

6.1.2 TM4L (TM for E-Learning)

TM4L je java aplikace pracující s TM4J API, vyvíjená na půdě Winston-Salem State University. Tato aplikace mě osobně velmi zklamala, nejen že je oproti OKS nepřehledná (user interface je podle mě naprosto nepoužitelný), ale během doby co jsem s TM4L pracovat nejméně 20 spadlo či přestalo reagovat. Nicméně v dokumentaci inzerují, že TM4l dokáže konvertovat v obou směrech, a to podle algoritmů [5] a [4]. Testovací soubor se mi v TM4L nepodařilo bohužel otevřít, přestože byl validní podle XTM specifikace. Konvertovat se mi po velkém počtu pokusů povedlo pouze mapy témat vytvořené v TM4L. Rozhodl jsem si vytvořit v TM4L stejnou testovací mapu jako v předchozím případě, kterou bych poté zkonvertoval, ale ani po třech pokusech se mi to nepodařilo, buď se mapa neuložila, vyběhla hláška typu `java.lang.nullPointerException`, nebo program nečekaně zamrzl. Nakonec se mi zázračně povedlo (jenom jednou až už nevím jak) zkonvertovat původní testovací mapu a toto je výsledek.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
     xmlns:xlink="http://www.w3.org/1999/xlink"/>
```

Myslím že TM4L ve verzi 2.0[38] je jako nástroj pro konverzi z TM do RDF nepoužitelný

6.2 Nástroje pro konverzi RDF do TM.

Když se nám podařilo data převést jedním směrem, mělo by to jít i opačnou cestou. Bohužel OKS nepodporuje importování RDF. Jediným nástrojem, který dokázal konvertovat RDF do TM je TM4L.

6.2.1 TM4L

TM4L převádí RDF do Topic Maps podle algoritmu [4]. Co mě velmi mile překvapilo je, že konverze tímto směrem běhá naprosto spolehlivě. Pro demonstraci jsem použil RDF tvrzení od Dave Becketta, která popisuje název, vydavatele a tvůrce Daveovi domácí stránky.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.ilrt.org/people/cmdjb/">
    <dc:title>Dave Beckett's Home Page</dc:title>
    <dc:creator>Dave Beckett</dc:creator>
    <dc:publisher>ILRT, University of Bristol</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Podívejme se jak se TM4L poradilo s těmito celkem jednoduchými RDF výroky. Velmi mě překvapila kvalita výsledného modelu, která přesně odpovídala RDF zdroji, rovněž se jmennými prostory pro vlastnosti `dc` si TM4L poradilo výborně. Jedinou chybičkou je využití `subjectIndicatorRef` namísto `resourceRef` u téma, reprezentující homepage, jedná se

[38] Aktuální verze ve chvíli psaní této práce

opravdu o drobnou chybičku jelikož konvertor nemůže podle URI poznat, zda se jedná o zdroj nebo předmět (a vzhledem k tomu,že v RDF jsou všechno zdroje...). Mrzutostí je fakt, že TM4L se neobtěžuje s nějakým zvýrazňováním syntaxe tabulátory nebo jinak, tedy výsledná mapa je po zkrašlovací kůře, kterou jsem provedl pro její přehlednost.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
     xmlns:xlink="http://www.w3.org/1999/xlink">

<topic id="id1">
     <subjectIdentity>
          <subjectIndicatorRef
          xlink:href="http://www.ilrt.org/people/cmdjb/"/>
     </subjectIdentity>
     <occurrence>
     <instanceOf>
          <topicRef xlink:href="#id2"/>
     </instanceOf>
          <resourceData>Dave Beckett's Home Page</resourceData>
     </occurrence>
     <occurrence>
     <instanceOf>
          <topicRef xlink:href="#id3"/>
     </instanceOf>
          <resourceData>Dave Beckett</resourceData>
     </occurrence>
     <occurrence>
          <instanceOf>
               <topicRef xlink:href="#id4"/>
          </instanceOf>
     <resourceData>ILRT, University of Bristol</resourceData>
     </occurrence>
</topic>

<topic id="id2">
     <subjectIdentity>
          <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/title"/>
     </subjectIdentity>
     <baseName>
          <baseNameString>dc:title</baseNameString>
     </baseName>
</topic>

<topic id="id3">
     <subjectIdentity>
          <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/creator"/>
     </subjectIdentity>
     <baseName>
          <baseNameString>dc:creator</baseNameString>
     </baseName>
</topic>

<topic id="id4">
     <subjectIdentity>
          <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/publisher"/>
     </subjectIdentity>
     <baseName>
          <baseNameString>dc:publisher</baseNameString>
```

```
    </baseName>
</topic>
</topicMap>
```

Pro úplnost jsem se pokusil konvertovat výsledek původní konverze (tedy OKS exportu TM do RDF), ovšem výsledný kód byl skoro delší než tato práce a bez zvýraznění struktury naprosto nepoužitelný pro jakékoliv demonstrační účely, takže se pouze pokusím slovně ohodnotit výsledek. Jak jsem předpokládal RDF reifikace reprezentující obor pravdivosti názvu v TM vztahu `ma-skladbu` se neposkládala od původního vztahu, ale vytvořila témata pro objekt, predikát `inscope` a subjekt reifikace a vztah mezi nimi, což je naprosto nepoužitelné. Tato chyba by šla napravit, pokud by TM4L bral v potaz, že `rdf:type` uzlu je `rdf:Statement`. Dalším podstatným háčkem bylo, že se mi tuto mapu témat bohužel nepodařilo otevřít v OKS, kvůli **syntaktické chybě XML**. Tato konverze tedy nesplnila základní předpoklad konverze a to interoperabilitu mezi aplikacemi. Myslím, že TM4L je jako nástroj pro transformaci jednoduchých RDF výroků postačující, ale k vyjádření složitějších konstrukcí a větších souborů dat určitě stačit nebude.

Další věc, kterou jsem se snažil převést, byl jednoduchý příklad OWLu, ale TM4L jej převedl jako obyčejné výroky, ani v dokumentaci nebyla zmínka o nějakém mapování těchto unikátních vlastností na možnosti TM.

7 Závěr

Co říci závěrem. Podle mého názoru a názoru L.M.Garshola [7] se **spojení obou technologií nejeví jako možné ani jako žádoucí**. Obě technologie mají **jiné možnosti určené pro jiné použití**. Každopádně se ukázalo jako **možné data** z jednotlivých technologií **převádět oběma směry** a to díky použití mapovaní vlastností specifických slovníků na jejich možné ekvivalenty v opačném formátu. Což umožňuje mimo samotné konverze i vzájemnému sdílení RDF a TM slovníků[7]. Dokonce i RDF a TM omezení jdou konvertovat oběma směry za použití specifického mapování, bohužel tato možnost zatím není v žádném konvertoru implementována. Sémantická anotace v OWLu může být přeložena přímo do mapy témat se stejným významem. Díky tomu, může být popisná část OWlu použita jak RDF a tak i samotnou technologií map témat. Stručně řečeno RDF a TM spolu mohou existovat bok po boku, aniž by jedna omezovala působnost té druhé. Zatím je před námi ještě dlouhý vývoj konvertorů, ale především specifických mapování vlastností obou technologií. Ovšem v současné době je tento vývoj zastíněn vývojem **dotazovacího jazyka TMQL**, který by měl být **schopen získávat informace** z obou formátů (**TM i RDF**), což bude pravděpodobně mít za následek velké zmenšení potřeby konverze dat[7].

Přínosem této práce je především stručné, ale úplné vyložení specifikací a možností obou technologií. Vysvětlení jejich použití a rozdílů mezi nimi. Dalším přínosem je vytvoření zprávy, která rozebírá současný stav nástrojů, které jsou schopny konvertovat data mezi formáty a hodnotí kvalitu transformací a výsledných modelu.

Výsledky této práce mohou být použitu v jakémkoliv výzkumu týkajícího se znalostního inženýrství a konkrétně pak TM, RDF, RDFS a OWL. Zajímavé téma pro návazné práce by mohlo být podrobné rozebrání návrhů a připravovaných možností dotazovacího jazyku TMQL.

8 Citovaná literatura

Citovaná literatura je řazena abecedně.

1. BARTA, Robert, GARSHOL, Lars Marius. *Topic Map Query Language (TMQL)* [online]. 2007 , 2007-10-31 [cit. 2007-12-20]. Dostupný z WWW: <http://www.isotopicmaps.org/tmql/>.
2. BECKETT, Dave. *N-Triples* [online]. 2007 [cit. 2007-12-07]. Dostupný z WWW: <http://www.w3.org/2001/sw/RDFCore/ntriples/>.
3. COVER, Robin. *Cover Pages: (XML) Topic Maps* [online]. 2005, August 13, 2005 [cit. 2007-12-20]. Dostupný z WWW: <http://xml.coverpages.org/topicMaps.html>.
4. DICHEVA, Darina, DICHEV, Christo. *TM4L - Topic Maps Editor for eLearning : RDF to Topic Maps Conversion* [online]. 2006 [cit. 2007-12-27]. Dostupný z WWW: <http://myweb.wssu.edu/dichevc/Topic-Maps/RDF-to-TM.html>.
5. DICHEVA, Darina, DICHEV, Christo. *TM4L - Topic Maps Editor for eLearning : Topic Maps to RDF Conversion* [online]. 2006 [cit. 2007-12-27]. Dostupný z WWW: <http://myweb.wssu.edu/dichevc/Topic-Maps/TM-to-RDF.html>.
6. GARSHOL, Lars Marius. *An RDF schema for topic maps* [online]. 2002 [cit. 2007-12-24]. Dostupný z WWW: <http://psi.ontopia.net/rdf/schema.rdf>. Popis schéma dostupný na <http://psi.ontopia.net/rdf/>.
7. GARSHOL, Lars Marius . *Living with topic maps and RDF* [online]. [2003] [cit. 2007-11-06]. Dostupný z WWW: <http://www.ontopia.net/topicmaps/materials/tmrdf.html>.
8. GARSHOL, Lars Marius. *RTM: An RDF-to-TM mapping* [online]. 2003 [cit. 2007-12-24]. Dostupný z WWW: <http://psi.ontopia.net/rdf2tm/schema.rdf>. Popis schema dostupný na <http://psi.ontopia.net/rdf2tm/>.
9. GARSHOL, Lars Marius. *The Linear Topic Map Notation* [online]. 2006, 2006/06/17 17:52:01 [cit. 2007-12-20]. Dostupný z WWW: <http://www.ontopia.net/download/ltm.html>.
10. GARSHOL, Lars Marius. *TMQL : Getting started* [online]. 2007 [cit. 2007-12-20]. Dostupný z WWW: <http://www.isotopicmaps.org/tmql/tmql-london-intro.pdf>.
11. FORTH institute of computer science. *The RDF Query Language (RQL)* [online]. 2003 , July 18, 2003 [cit. 2007-12-21]. Dostupný z WWW: <http://139.91.183.30:9090/RDF/RQL/>.
12. KOSEK, Jiří. *Topic Maps : Tutoriál* [online]. 2006 [cit. 2007-12-14]. Dostupný z WWW: <http://www.kosek.cz/xml/tmtut/>.
13. Ontopia. *The Ontopia Schema Language : Tutorial* [online]. 2007, 2007/08/24 14:44:07 [cit. 2007-12-19]. Dostupný z WWW: <http://www.ontopia.net/omnigator/docs/schema/tutorial.html>.
14. Ontopia A/S. *The RTM RDF to topic maps mapping : Definition and introduction* [online]. 2003 [cit. 2007-11-11]. Dostupný z WWW: <http://www.ontopia.net/topicmaps/materials/rdf2tm.html>.
15. Ontopia. *Tolog for TMQL?* [online]. 2007 [cit. 2007-12-20]. Dostupný z WWW: <http://www.isotopicmaps.org/tmql/tolog.pdf>.
16. PEPER, Steve. *Ontopia: Topic Maps and RDF: Ten Theses on Topic Maps and RDF* [online]. 2002 [cit. 2007-12-20]. Dostupný z WWW: <http://www.ontopia.net/topicmaps/materials/rdf.html>.
17. PETRÁK, Josef, ZEMÁNEK, Jan. *RDF: Často kladené dotazy* [online]. 2003, 30. listopadu 2003 [cit. 2007-12-14]. Dostupný z WWW: <http://dsic.zapisky.info/archive/RDF-FAQ-20031130/>.
18. SEABORNE, Andy. *Jena Tutorial - RDQL* [online]. 2002, February 2004 [cit. 2007-12-21]. Dostupný z WWW: <http://jena.sourceforge.net/tutorial/RDQL/>.
19. SeSame. *User Guide for Sesame : Chapter 6. The SeRQL query language (revision 1.2)* [online]. 2006 [cit. 2008-01-01]. Dostupný z WWW: <http://www.openrdf.org/doc/sesame/users/ch06.html#d0e1056>.

20. SVÁTEK, Vojtěch. *Ontologie a WWW* [online]. 2003 [cit. 2007-11-01]. Dostupný z WWW: <http://nb.vse.cz/~svatek/onto-www.pdf>.
21. ŠVIHLA, Martin. *Uložení a dotazování RDF* [online]. 2007 [cit. 2007-12-21]. Dostupný z WWW: <http://nb.vse.cz/~svatek/rzzw/RDFstorage.pdf>.
22. ŠVIHLA, Martin. *Úvod do RDF a RDF-S* [online]. 2007 [cit. 2007-12-07]. Dostupný z WWW: <http://nb.vse.cz/~svatek/rzzw/RDF.pdf>. http://semestr.vse.cz (pro studenty VŠE, IZ440).
23. TopicMaps.Org Authoring Group. *XML Topic Maps (XTM) 1.0* [online]. 2001 [cit. 2007-11-01]. Dostupný z WWW: <http://www.topicmaps.org/xtm/>.
24. W3C. *Archive of W3C News* [online]. 2007, 2007/12/20 17:22:58 [cit. 2007-12-21]. Dostupný z WWW: <http://www.w3.org/News/2007#item247>.
25. W3C. *OWL Web Ontology Language Overview* [online]. 2004 [cit. 2008-01-01]. Dostupný z WWW: <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1>.
26. Wikipedia. *Notation 3* [online]. 2007 [cit. 2007-12-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Notation_3>.
27. Wikipedia. *RDF Schema* [online]. 2007 [cit. 2007-12-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/RDF_Schema>.
28. Wikipedia. *Resource Description Framework* [online]. 2007 [cit. 2007-12-06]. Angličtina. Dostupný z WWW: <http://en.wikipedia.org/wiki/Resource_Description_Framework>.
29. Wikipedia. *SPARQL* [online]. 2007 [cit. 2007-12-21]. Dostupný z WWW: <http://en.wikipedia.org/wiki/SPARQL>.
30. Wikipedia. *Turtle (syntax)* [online]. 2007 [cit. 2007-12-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Turtle_%28syntax%29>.
31. Wikipedia. *Web Ontology Language* [online]. 2007 [cit. 2007-12-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Web_Ontology_Language>.

9 Seznam použitých zkratek

Toto je se seznam frekventovaně užívaných termínů, a jejich definic

- HTML (HyprText Markup Language) – Hypertextový značkovací jazyk, který se
- HyTime (Hypermedia/Time-based Structurink Language) - značkovací jazyk založený na SGML. http://en.wikipedia.org/wiki/HyTime
- OKS (Ontopia Knowledge Suite) – Kompletní balík nástrojů pro práci s TM od firmy Ontopia (komerční) více na http://www.ontopia.net/solutions/products.html
- Ontologie – sdílená formální explicitní konceptualizace. Více v [19]
- OQL (Object Query Language) – Dotazovací jazyk pro objektově orientované databáze, vyvinuto společností Object Data Management Group (ODMG) http://en.wikipedia.org/wiki/Object_Query_Language
- OWL (Ontology Web Language) – jazyk pro webové ontologie více v kapitole (2.2.1) a na http://en.wikipedia.org/wiki/Web_Ontology_Language
- RDF (Resource Description Framework) – rámec pro popis zdrojů viz kapitola 2.
- RDF schema (RDF-S) - více v kapitole (2.2.1).
- SAM (The Standard Application Model for Topic Maps) – standardní aplikační model pro TM http://www.isotopicmaps.org/sam/sam-model/
- TM (Topic Maps) - mapy témat viz kapitola 3.
- URI (Uniform Resource Identifier) – Jednotný identifikátor zdroje
- URL (Uniform Resource Locator) – Jednotný lokátor zdroje – podmnožina URI
- W3C (Word Wide Web Consorcium) – konsorcium pro vývoj WWW http://www.w3c.org/
- WWW (World Wide Web) - používá k tvorbě www dokumentů.
- XML (eXtensible Markup Language) – rozšiřitelný značkovací jazyk. http://www.w3.org/XML

10 Přílohy

K této práci nejsou žádné přílohy přiloženy.