

Reaktivní ontologické inženýrství (na sémantickém webu)

Vojtěch Svátek

*Katedra informačního a znalostního inženýrství
Vysoká škola ekonomická v Praze*



LAAOS

Pro informaci: Prezentace byla původně připravena pro zvanou přednášku na konferenci *Znalosti 2013*.
Následně byla upravena pro předmět Umělá inteligence na UHK a předmět Propojená data na webu na VŠE.

Osnova

- Vývoj praxe tvorby ontologií
(na sémantickém webu i mimo něj)
- Linked data a datové slovníky pro ně
- Reaktivní ontologické inženýrství (ROI)
- Vybrané projekty ROI pro linked data
 - PatOMat – transformace ontologie/slovníku
 - ORE – korekce a obohacování slovníku s využitím dat
 - PURO – ontologické pozadí slovníku
- Shrnutí

Tvorba ontologií

- 90. léta – vznikají rozsáhlé doménové ontologie
 - Tvoří je týmy znalostních inženýrů a expertů
 - Využití rozsáhlých podkladových materiálů, komplexní pokrytí zvolené domény
 - Od seznamu termínů přes výkladový glosář, formalizaci v bohatém jazyce (minimálně 1. řádu), ke kódování v jazyce pro odvozování (např. OCML, deskripční logika, ev. jazyky typu Lisp nebo Prolog)
 - Formalizována je často jen část ontologie
 - Přepoužívané širší, dobře vymezenou doménovou komunitou, jako celek nebo ve větších modulech
 - Např. Enterprise Ontology, GALEN

Tvorba ontologií

- Proliferace „webových ontologií“ po roce 2000
 - Jazyk OWL, resp. jeho předchůdci
 - Často jen vágní grafické schéma, a pak hned přímé kódování v editoru (Protégé, TopBraid, NeOn, ...)
 - Nesystematicky zařazovány různé typy axiomů, někdy na úkor shody s realitou (nejsou ošetřeny významné výjimky)
 - Ontologie, ani jednotlivé entity z nich, často nejsou reálně používány nikde („ontologie pro ontologii“), nebo jen v aplikaci, pro kterou vznikly
- Reálně úspěšné „ontologické“ projekty v praxi spíš mimo hlavní proud sémantického webu

Tvorba ontologií

- Nástup datových slovníků v polovině 1. desetiletí
 - Malý rozsah a jednoduchá struktura: jedna nebo několik hlavních tříd, širší množina vlastností, někdy s nejasně určeným oborem hodnot
 - Určené pro rozsáhlé distribuované komunity pracující s RDF, někdy vznik v rámci „vocampů“
 - Některé poměrně široce využívané
 - Dublin Core – obecná metadata „publikací“
 - FOAF – informace o lidech a jejich vzájemných vztazích
 - SKOS – tezaurové taxonomie pojmů
 - SIOC – účast v online fórech
 - DOAP – open source softwarové projekty
 - GoodRelations – nabídky obchodních transakcí
 - Míra využívání jednotlivých entit ze stejného slovníku se může velmi lišit

Tvorba ontologií - současnost

1. Rozsáhlé **doménové** ontologie

- nadále vznikají zejména v biomedicíně (viz <http://bioportal.bioontology.org/>), např. ICD (Mezinárodní klasifikace nemocí)
- unifikovaná struktura (např. jen existenční restriktce - OWL EL), generovaná automaticky pomocí šablon
- příklad ze SNOMED-OWL (v Manch. syntaxi)

```
Appendicectomy equivalentTo  
Surgical_Procedure and  
method some Excision) and  
procedure-site some Appendix_structure)
```

Tvorba ontologií - současnost

2. Ontologie pro lepší přístup ke konkrétnímu datovému zdroji - tzv. **OBDA**
 - kompaktní, využívané aplikací jako celek, využívání přinejmenším zčásti na základě CWA
 - spíš nadstavba relačního datového schématu než skutečná „ontologie sémantického webu“
3. Datové slovníky pro **linked data**
 - zajišťují minimální vyjádření „významu“ dat v infrastruktuře bez vymezeného cílového určení
 - provázané mezi sebou, kombinují se

Linked (open) data

- Data vystavená ve formátu **RDF** na **webu**
 - Trojice „subjekt – predikát – objekt“
 - Praha *type* City
 - Praha *country* CzechRepublic
 - Praha *hasPopulation* „1 250 000“
 - Téměř vždy ale původně vznikla v jiném formátu
- Entity („zdroje“, „URI“) z různých datasetů **propojeny** identitními (*owl:sameAs*) aj. linky
 - *vse:ondrej_zamazal owl:sameAs dblp:ondrej_svab_zamazal*
 - *kizi:patomat dcterms:creator dblp:ondrej_svab_zamazal*

Vystavení linked data na webu

- „Levné“ způsoby vystavení:
 - **dumpy** trojic (N-Triples) ke stažení
 - **RDFa** - zanoření trojic RDF do HTML
- „Drahé“ způsoby vystavení:
 - **linked data** v užším smyslu: poskytování popisů zdrojů při *dereferencování* URI, *negociace obsahu*, navigace přes odkazy (linky) na další zdroje
 - koncové body **SPARQL** (webová API, dotazování bohatým jazykem podobným SQL)

Navigace linked data v prohlížeči

About: [Prague](#)

An Entity of Type : [populated place](#), from Named Graph : <http://dbpedia.org>, within Data Space : dbpedia.org



Prague is the capital and largest city of the Czech Republic. Situated in the north-west of the country on the Vltava river, the city is home to about 1.3 million people, while its metropolitan area is estimated to have a population of over 2.3 million. The city has a temperate oceanic climate with warm summers and chilly winters. Prague has been a political, cultural and economic centre of Europe and particularly central Europe during its 1,100 year existence.

| | |
|--|---|
| dbpedia-owl:areaTotal | ▪ 496000000.000000 (xsd:double) |
| dbpedia-owl:country | ▪ dbpedia:Czech_Republic |
| dbpedia-owl:leaderName | ▪ dbpedia:Bohuslav_Svoboda |
| dbpedia-owl:leaderParty | ▪ dbpedia:Civic_Democratic_Party_(Czech_Republic) |
| dbpedia-owl:leaderTitle | ▪ Mayor |
| dbpedia-owl:maximumElevation | ▪ 399.000000 (xsd:double) |
| dbpedia-owl:motto | ▪ (Prague, Head of the State; Latin) |
| dbpedia-owl:populationAsOf | ▪ 2011-01-14 (xsd:date) |
| dbpedia-owl:populationMetro | ▪ 2300000 (xsd:integer) |
| dbpedia-owl:populationTotal | ▪ 1290846 (xsd:integer) |
| dbpedia-owl:postalCode | ▪ 1x xx |
| dbpedia-owl:thumbnail | ▪ http://upload.wikimedia.org/wikipedia/commons/thumb/ |
| dbpedia-owl:timeZone | ▪ dbpedia:Central_European_Time |

Slovníky linked data

- Aktuálně sdruženy v katalogu **Linked Open Vocabularies** (LOV) spravovaným OKFN
 - <http://lov.okfn.org/dataset/lov/>
 - Aktuálně přes 400 „slovníků“
- Kromě typických slovníků LD zahrnuje i rozsáhlejší taxonomické klasifikace a axiomatizované ontologie
- Mapuje různé typy přepoužití mezi slovníky a sleduje časový vývoj verzí

Reaktivní ontologické inženýrství

- Reaktivní aspekt byl od začátku přítomný ve smyslu transformace „ne-ontologických“ zdrojů (tj. nevyjádřených v patřičném logickém formalismu) na „ontologické“
 - Využití tezaurů, podnikových taxonomií, produktových klasifikací, zdravotnických nomenklatur, UML diagramů, relačních datových schémat, apod.
- Nová je možnost masové aplikace reaktivního přístupu na zdroje, které už *syntakticky* v daném formalismu (OWL/RDFS) jsou, ale jsou po strukturní nebo lexikální stránce *neúplné* nebo *nepřesné*
 - Zejména slovníky linked data

Možné „reakce“ na stav ontologie

- **Reakce 1: Změna struktury** – vytvoření „varianty“ ontologie s jiným „stylem modelování“
→ hlavní směr projektu **PatOMat**
- **Reakce 2: Přidávání/odstraňování axiomů**
(při zachování základní struktury)
→ (mj.) projekt **ORE**
- **Reakce 3: Doplnění potřebných informací do samostatné nadstavby**, která je s ontologií propojena
→ (mj.) *modely pozadí*, jako je **PURO**

Reakce 1: Projekt PatOMat

„Let’s make ontologies metamorphic“

- Metamorfóza v přírodě: jde o stejného jedince, ale vypadá jinak
- Metamorfóza ontologií: realita na pozadí dané ontologie je stejná, ale styl jejího modelování navenek se změní



Základní motivace projektu

- Jazyk OWL umožňuje vyjádřit stejný reálný „stav věcí“ různými formálními konstrukcemi, které můžeme chápat jako různý „styl modelování“
- Ontologické struktury odpovídající různým stylovým variantám lze popsat pomocí **vzorů** („patterns“ – z toho **PatOMat**)
- Fragmenty ontologií lze na sebe automaticky **transformovat** tak, aby stále odrážely stejný reálný „stav věcí“

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

accepts Domain: PCChair. accepts Range: Paper.

rejects Domain: PCChair. rejects Range: Paper.

accepts DisjointWith: rejects.

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

accepts Domain: PCChair. accepts Range: Paper.

rejects Domain: PCChair. rejects Range: Paper.

accepts DisjointWith: rejects.

hasPCChairDecision Domain: Paper.

hasPCChairDecision Range: (EquivalentTo {acceptance, rejection}).

hasPCChairDecision Characteristics: FunctionalProperty.

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

accepts Domain: PCChair. accepts Range: Paper.

rejects Domain: PCChair. rejects Range: Paper.

accepts DisjointWith: rejects.

hasPCChairDecision Domain: Paper.

hasPCChairDecision Range: (EquivalentTo {acceptance, rejection}).

hasPCChairDecision Characteristics: FunctionalProperty.

hasPCChairDecision Domain: Paper

hasPCChairDecision Range: Decision.

Acceptance SubClassOf: Decision.

Rejection SubClassOf: Decision.

Acceptance DisjointWith: Rejection.

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

accepts Domain: PCChair. accepts Range: Paper.

rejects Domain: PCChair. rejects Range: Paper.

accepts DisjointWith: rejects.

hasPCChairDecision Domain: Paper.

hasPCChairDecision Range: (EquivalentTo {acceptance, rejection}).

hasPCChairDecision Characteristics: FunctionalProperty.

hasPCChairDecision Domain: Paper

hasPCChairDecision Range: Decision.

Acceptance SubClassOf: Decision.

Rejection SubClassOf: Decision.

Acceptance DisjointWith: Rejection.

Sémantika vzniká v souhře
logické struktury
a pojmenování entit

Příklad „stylových“ variant

PaperAcceptedByPCChair SubClassOf: Paper.

PaperRejectedByPCChair SubClassOf: Paper.

PaperAcceptedByPCChair DisjointWith: PaperRejectedByPCChair.

accepts Domain: PCChair. accepts Range: Paper.

rejects Domain: PCChair. rejects Range: Paper.

accepts DisjointWith: rejects.

hasPCChairDecision Domain: Paper.

hasPCChairDecision Range: (EquivalentTo {acceptance, rejection}).

hasPCChairDecision Characteristics: FunctionalProperty.

hasPCChairDecision Domain: Paper

hasPCChairDecision Range: Decision.

Acceptance SubClassOf: Decision.

Rejection SubClassOf: Decision.

Acceptance DisjointWith: Rejection.

Další varianty vznikají při náhradě objektových vlastností **datovými**: s hodnotami string, boolean, atd. místo ontologických instancí

Softwarové nástroje PatOMat

- **Framework** provádějící detekce a transformace vzorů
 - jako RESTovské služby nebo knihovna Java
 - 3 fáze: detekce výskytů, generování transformačních instrukcí (pro vybrané instance vzorů), vlastní transformace
- **Editory** transformačních vzorů *TPE* a *WEEdOP*
- **Generická** rozhraní pro **provádění transformací**:
 - **grafické rozhraní** *GUIPOT* (plugin do Protégé)
 - **volání služeb** z portálu *WebCOP*
- Rozhraní pro provádění **specifických** typů transformací
 - Mj. pro nástroje **XDtools** (CNR Řím) a **ORE** (Univ. Lipsko)

Web projektu PatOMat

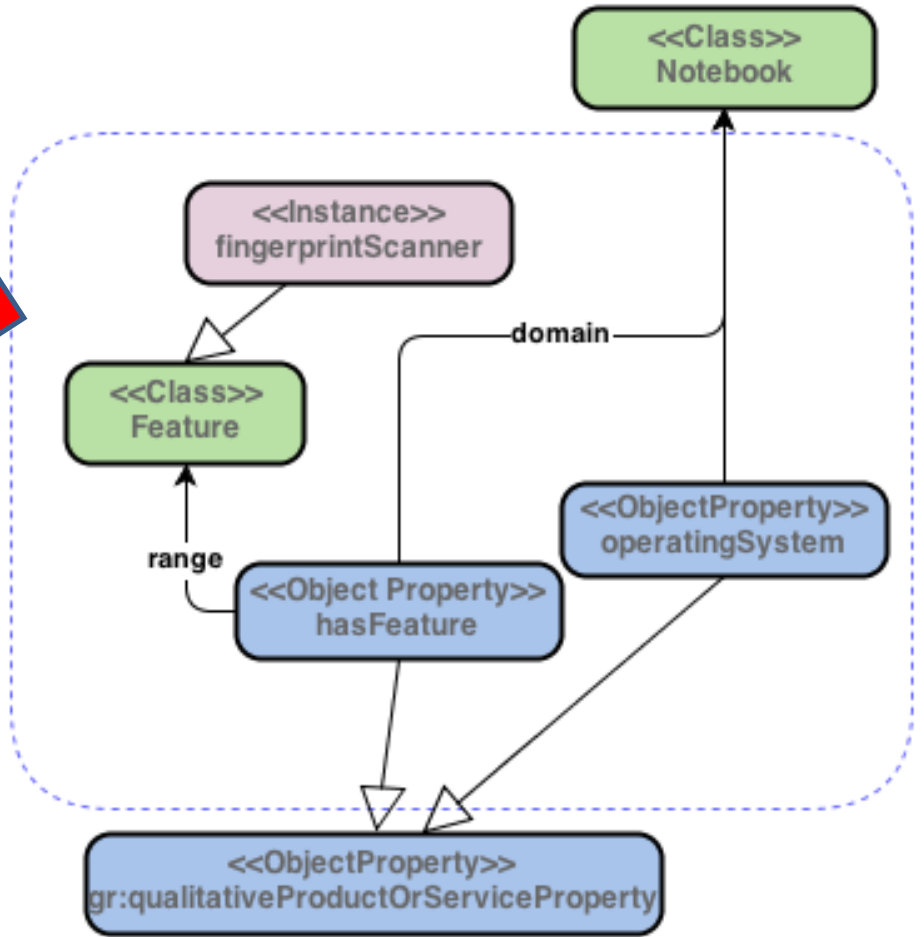
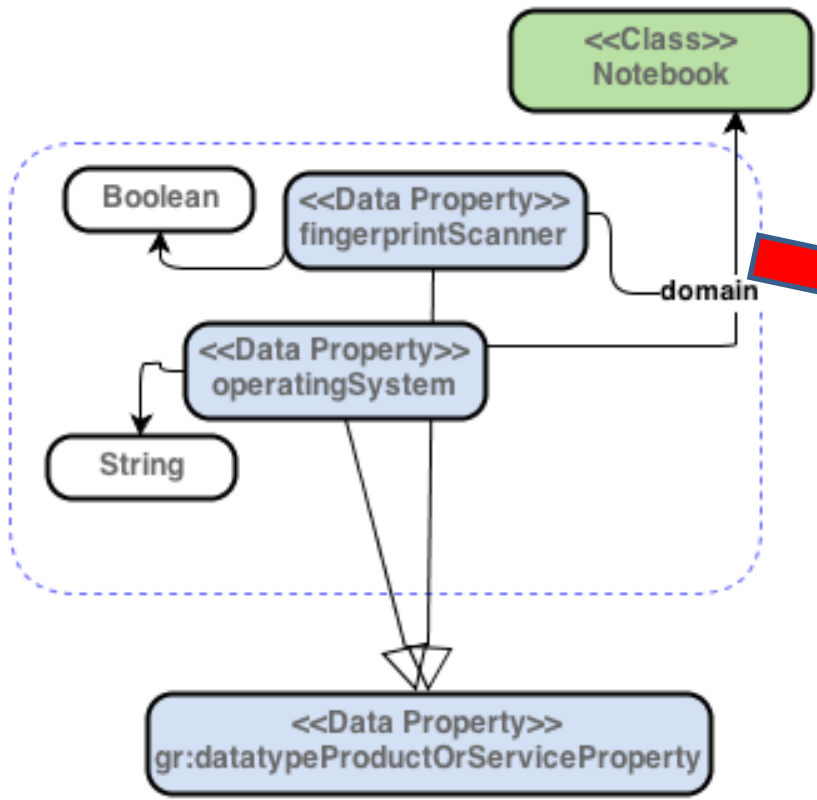
- <http://patomat.vse.cz> – souhrnné informace o projektu, případy užití, odkazy na články
- <http://owl.vse.cz:8080> – odkazy na nástroje a jiné artefakty, online tutoriály apod.

Příklad aplikace

- V oblasti e-commerce vznikají ontologie určené pro výstřižky o **produktech** a **službách** na webových stránkách (RDFa, mikrodata) – např. projekt **OPDM**
- Pro snadnou tvorbu šablon webmastery a zpracování vyhledávači jsou zde preferovány **datové vlastnosti** (hodnoty = literály)
- Pro druhotné využití ontologií v rámci **doporučovacích** systémů by ale byly vhodnější **objektové vlastnosti** (hodnoty = URI)
 - předpoklad propojování na externí linked data
- Případová studie doporučovacího systému: párování nabídky a poptávky na trhu **veřejných zakázek**
- Pro využití v „párovací“ aplikaci (projekt EU LOD2) bylo proto třeba ontologie **transformovat**

E-commerce na webu

Doporučování, párování



Na úrovni trojic RDF

pc:laptopA1 pc:fingerprintScanner true.

pc:laptopA1 pc:operatingSystem "Win7".



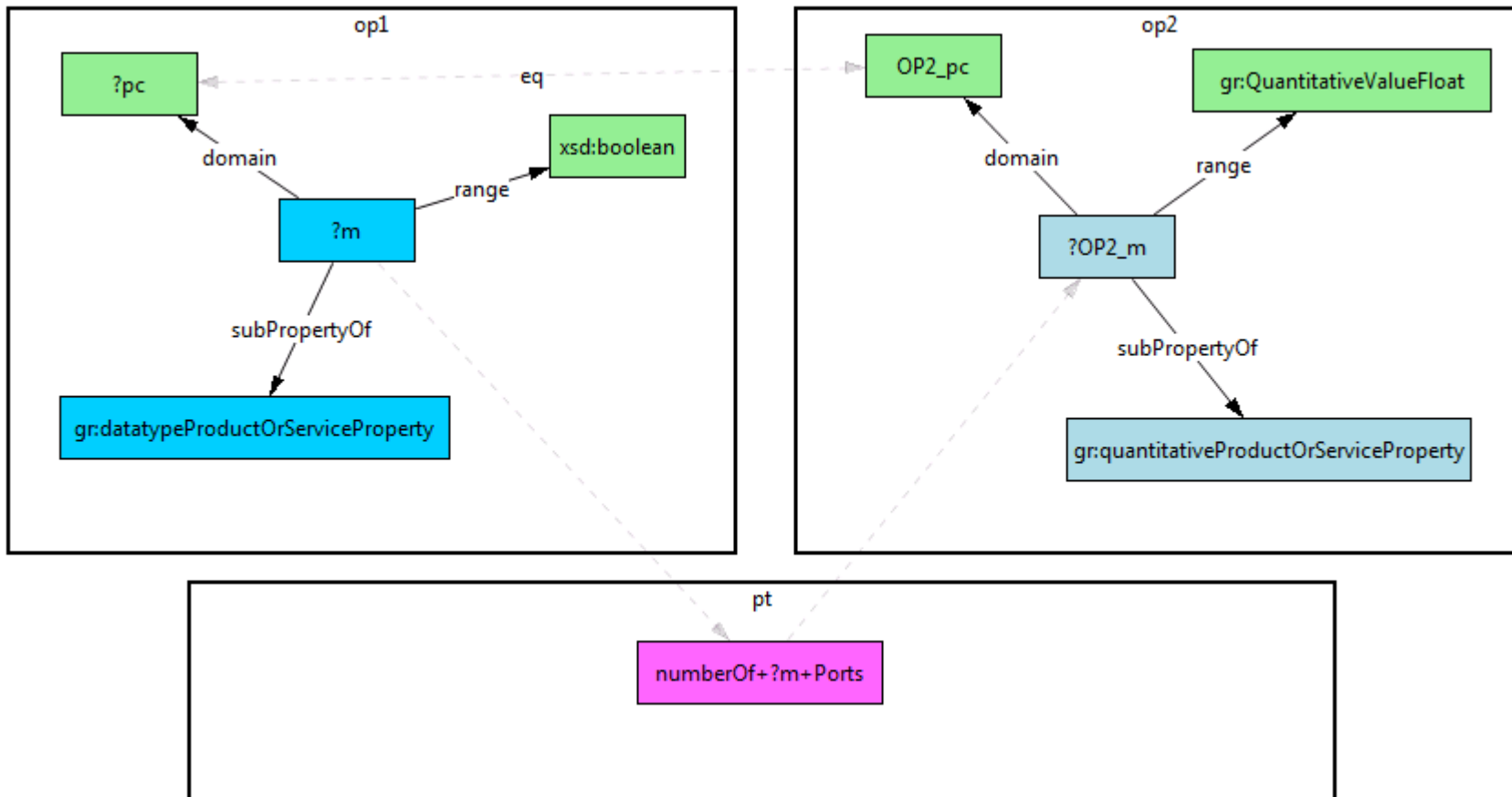
pc:laptopA1 pc:hasFeature pc:fingerprintScanner.

pc:laptopA1 pc:operatingSystem pto:Windows_7.

Propojení na další zdroj v rámci linked data:

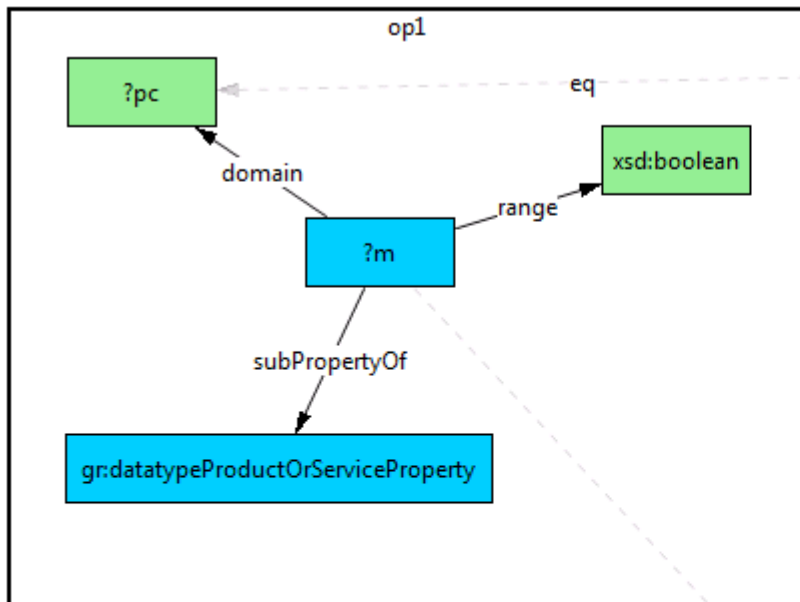
pc:fingerPrintScanner rdfs:seeAlso dbpedia:Fingerprint_scanner

Transformační vzor (v editoru)

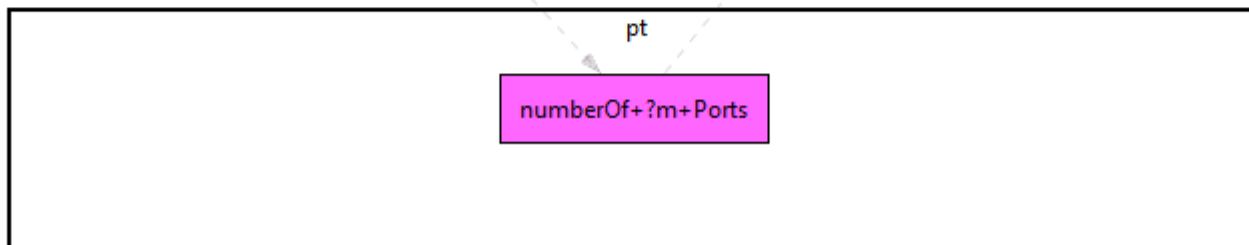
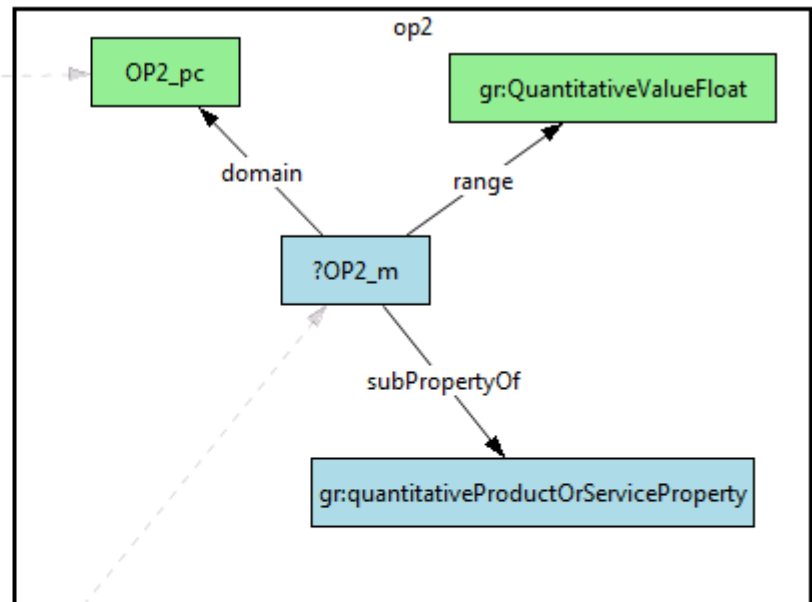


Transformační vzor (v editoru)

Vstupní vzor



Výstupní vzor



Transformační propojení + transformace pojmenování entit

Transformace v nástroji GUIPOT

The screenshot displays the GUIPOT application interface, which is used for applying transformations to a knowledge base. The interface is divided into several main sections:

- Left Panel (Tree View):** Shows a hierarchical tree of classes. The root is "Thing", which branches into "ProductOrService", "DeliveryMethod", "MetadataProfile", "WarrantyPromise", "Offering", and "Product". Under "Product", the class "PortableMediaPlayer" is highlighted in green and labeled with "?pc?pc". Other classes include "SomeItems", "ProductOrServiceModel", "ProductOrServicesSomeInstancesPl", "Individual", "ActualProductOrService", "OpeningHoursSpecification", "PriceSpecification", "MetadataContainer", "PaymentMethod", "Brand", "QuantitativeValue", "BusinessEntityType", "BusinessFunction", "LocationOfSalesOrServiceProvisioning", and "TypeAndQuantityNode".
- Top Panel (Configuration):** Contains a "Load TP" field with the file path "file:/C:/Users/Marek/Documents/ecweb2013/bool_inst.xml" and a "Choose TP file" button. There is a checkbox for "Use recursive detection" and a dropdown menu for "strategy" set to "conservative". A large "Apply transformations" button is present.
- Middle Panel (Transformation Rules):** A list of transformation rules, each starting with "?pc=PortableMediaPlayer;?m=" followed by a specific media format or feature. The selected rule is "?pc=PortableMediaPlayer;?m=mp3". Other rules include "integratedCamera", "touchscreen", "integratedMicrophone", "mpeg2", "memoryStick", "changeableBattery", "dab", "divX", "wmv", "bluetooth", "avi", and "mpeg4".
- Bottom Panel (Transformed Objects):** Shows the results of the transformation. The left side lists objects like "integratedCamera", "mp3", "voiceRecording", "storageMedium", "bluetooth", "gpcBrick", "eClass", "wlan", "avi", "usb", "fmRadio", and "directRecordingFromTV". The right side shows a list of classes including "BusinessEntityType", "BusinessFunction", "Category", "CategoryScheme", "DayOfWeek", "DeliveryMethod", "DeliveryModeParcelService", "MetadataProfile", "PaymentMethod", "PaymentMethodCreditCard", "PortableMediaPlayer", "Various_iPods.jpg", and "WarrantyScope".
- Right Panel (Output):** A simple view showing the class "Thing" and a "topObjectProperty" label.

Transformace v nástroji GUIPOT

The screenshot displays the GUIPOT (Graphical User Interface for Protégé) software. The main window shows a list of transformation rules on the left, a central control panel, and a right-hand pane showing the resulting ontology structure.

Transformation Rules List:

- ?pc=PortableMediaPlayer;?m=voiceRecording
- ?pc=PortableMediaPlayer;?m=mp3
- ?pc=PortableMediaPlayer;?m=integratedCamera
- ?pc=PortableMediaPlayer;?m=touchscreen
- ?pc=PortableMediaPlayer;?m=integratedMicrophone
- ?pc=PortableMediaPlayer;?m=mpeg2
- ?pc=PortableMediaPlayer;?m=memoryStick
- ?pc=PortableMediaPlayer;?m=changeableBattery
- ?pc=PortableMediaPlayer;?m=dab
- ?pc=PortableMediaPlayer;?m=divX
- ?pc=PortableMediaPlayer;?m=wmv
- ?pc=PortableMediaPlayer;?m=bluetooth
- ?pc=PortableMediaPlayer;?m=avi

Control Panel:

- Load TP: file://C:/Users/Marek/Documents/ecweb2013/bool_inst.xml
- Use recursive detection:
- strategy: radical-remove
- Apply transformations
- Show transformation links:
- Load result to Protégé
- Semantic zoom: + - Stop Motion

Resulting Ontology Structure (Right Pane):

- Place ≡ LocationOfSalesOrServiceProv
- WarrantyScope
- QualitativeValue <-
 - CategoryScheme
 - FeatureGroup
 - MediaFormat
 - Category
- DayOfWeek
- DeliveryMethod
- MetadataProfile
- Offering
- WarrantyPromise
- ProductOrService ≡ Product
 - PortableMediaPlayer <-
 - SomeItems ≡ ProductOrServicesSo
 - ProductOrServiceModel
 - ProductOrServicesSomeInstancesP
 - Individual ≡ ActualProductOrService
 - ActualProductOrServiceInstance ≡
 - OpeningHoursSpecification
 - MetadataContainer
 - Brand
 - QuantitativeValue

Transformation Results (Bottom Panes):

- datatypeProductOrServiceP:** integratedCamera, voiceRecording, bluetooth, storageMedium, gpcBrick, eClass, wlan, avi ?m, usb, fmRadio, directRecordingFromTV, memoryStick
- BusinessEntityType:** BusinessEntityType, BusinessFunction, Category, CategoryScheme, DayOfWeek, DeliveryMethod, DeliveryModeParcelSer, MediaFormat, mp3 <-, avi <-, MetadataProfile, PaymentMethod, PaymentMethodCredit
- topObjectProperty:** lesser, greater, qualitativeProductOrServicePropert
 - hasManufacturer
 - playbackFormat <-hasBrand, lesserOrEqual, hasOpeningHoursSpecification, availableDeliveryMethods

Další aplikace / případy užití

- Převod mezi hierarchií (tříd) OWL a (instancí) SKOS
- Snížení logické complexity ontologie
- *Doplnění neúplného pojmenování entit (viz **ORE**)*
- Usnadnění automatického mapování konceptů mezi ontologiemi
- Umožnění integrace ontologií (ev. obsahových vzorů)
- Přehlednější vizualizace ontologie

Reakce 2: Projekt ORE

- „Ontology Repair and Enrichment“
- Projekt řešený na Univ. Lipsko cca od r. 2009
<http://ore-tool.net/Projects/ORE>, <http://ore.aksw.org/ore/>
- **Korekce** („repair“) ontologií je široce zkoumaným problémem, zde se však nově
 - neomezuje na logickou konzistenci a strukturní pravidelnosti, ale pracuje i s **pojmenováním** entit – zde využití *PatOMat*
 - aplikuje na slovníky linked data, porovnává schéma s **instancemi**
- **Obohacování** ontologií/slovníků o nové axiomy na základě linked data bylo v době vzniku projektu průkopnickým výzkumem
 - přímé napojení na **koncové body SPARQL** – „proudová“ analýza linked data v reálném čase
- Původně desktopová aplikace, v současnosti je primárně rozvíjena webová verze

„Naming problem“ v ontologii

- Může indikovat konceptuální chybu, neúplnost pojmenování, nebo být jen „falešným poplachem“ plynoucím z nedostatečného pokrytí terminologickým zdrojem při analýze
 - PCChair SubClassOf: ProgramCommittee.
 - Accepted SubClassOf: Paper.
 - InvitedTalk SubClassOf: Presentation.

„Naming problem“ v ontologii

- Může indikovat konceptuální chybu, neúplnost pojmenování, nebo být jen „falešným poplachem“ plynoucím z nedostatečného pokrytí terminologickým zdrojem při analýze
 - PCChair SubClassOf: ProgramCommittee.
 - Accepted SubClassOf: Paper.
 - InvitedTalk SubClassOf: Presentation.

„Naming problem“ v ontologii

- Může indikovat konceptuální chybu, neúplnost pojmenování, nebo být jen „falešným poplachem“ plynoucím z nedostatečného pokrytí terminologickým zdrojem při analýze
 - PCChair SubClassOf: ProgramCommittee.
 - Accepted SubClassOf: Paper.
 - InvitedTalk SubClassOf: Presentation.

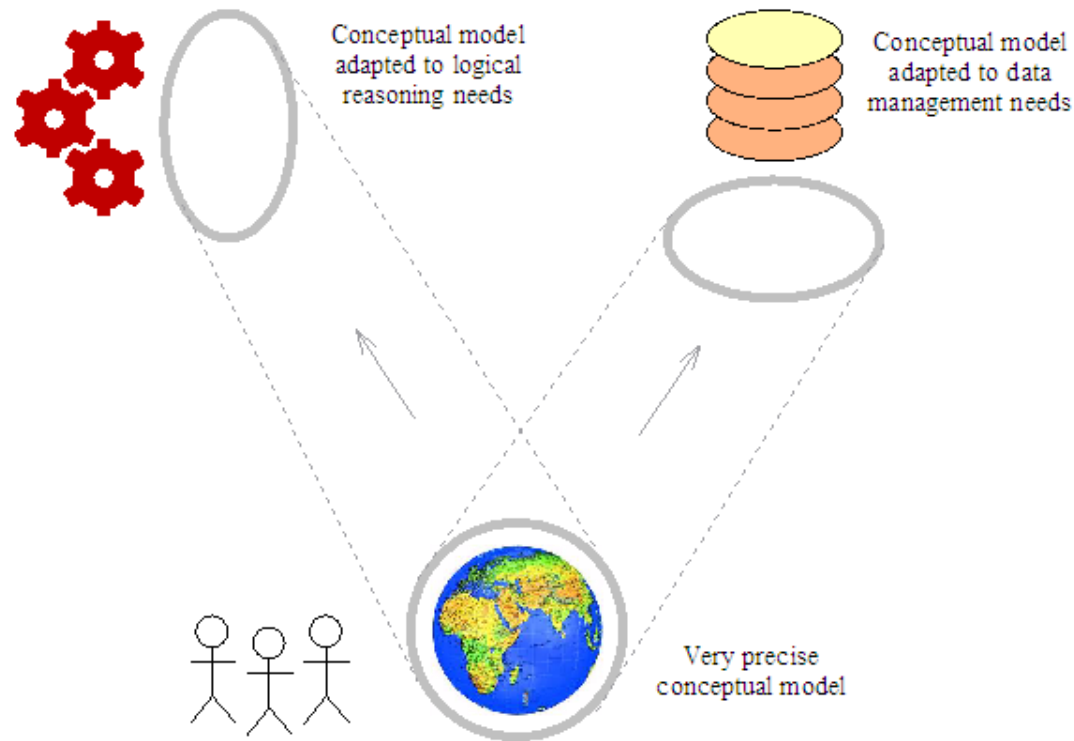
„Naming problem“ v ontologii

- Může indikovat konceptuální chybu, neúplnost pojmenování, nebo být jen „falešným poplachem“ plynoucím z nedostatečného pokrytí terminologickým zdrojem při analýze
 - PCChair SubClassOf: ProgramCommittee.
 - Accepted SubClassOf: Paper.
 - InvitedTalk SubClassOf: Presentation.

Reakce 3: Modely ontologického pozadí

- Ontologické inženýrství by na rozdíl od databázového / softwarového inženýrství mělo aspirovat na zachycení reality „tak, jak je“, relativně nezávisle na požadavcích databází a konkrétních softwarových systémů
 - Už název „ontologický“ je zavazující!
- Jak moc ale toto dnes platí?

Konceptualizace, logika a management dat



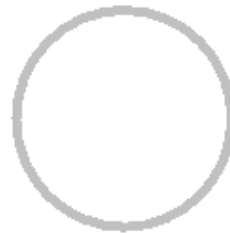
Konceptualizace, logika a management dat



Conceptual model
adapted to logical
reasoning needs




Conceptual model
adapted to data
management needs



Very precise
conceptual model

Konceptualizace, logika a management dat



Conceptual model
adapted to logical
reasoning needs

Conceptual model
adapted to data
management needs

Modely ontologického popředí
(ontological foreground models, OFMs)

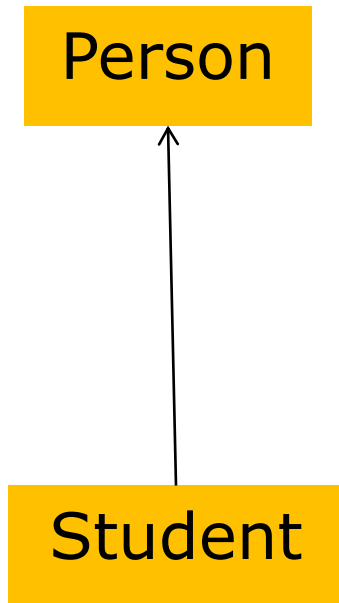
Modely ontologického pozadí
(ontological background model, OBM)

Very precise
conceptual model

Jazyky pro OFM a OBM

- Oba druhy modelů musí být vyjádřeny v určitém jazyce, obsahujícím potřebná reprezentační primitiva
- Obvyklým jazykem pro modely popředí (OFML) je v prostředí sémantického webu **OWL**
 - Zahrnuje v sobě jednodušší jazyky používané pro linked data, jako je RDFS
- Jak může vypadat jazyk pro modely pozadí (OBML)?
 - Existujícím příkladem je **OntoClean**, viz [samostatná prezentace](#)

OntoClean: OBML pro taxonomicky zaměřené ontologie



- Rigidní třída
- Má vlastní kritérium identity
- Má kritérium jednoty

- Antirigidní třída
- Nemá kritérium identity
- Má kritérium jednoty

OntoClean: OBML pro taxonomicky zaměřené ontologie

Anotování „meta-vlastnostmi“

+R+O+I+U

Person

- Rigidní třída
- Má vlastní kritérium identity
- Má kritérium jednoty

~R-O-I+U

Student

- Antirigidní třída
- Nemá kritérium identity
- Má kritérium jednoty

OntoClean: OBML pro taxonomicky zaměřené ontologie

Anotování „meta-vlastnostmi“

+R+O+I+U

Person

- Rigidní třída
- Má vlastní kritérium identity
- Má kritérium jednoty

~R-O-I+U

Student

- Antirigidní třída
- Nemá kritérium identity
- Má kritérium jednoty

Možnost testovat koherenci, např. $P^{\sim R}$ nemůže být nadřazeno Q^{+R}

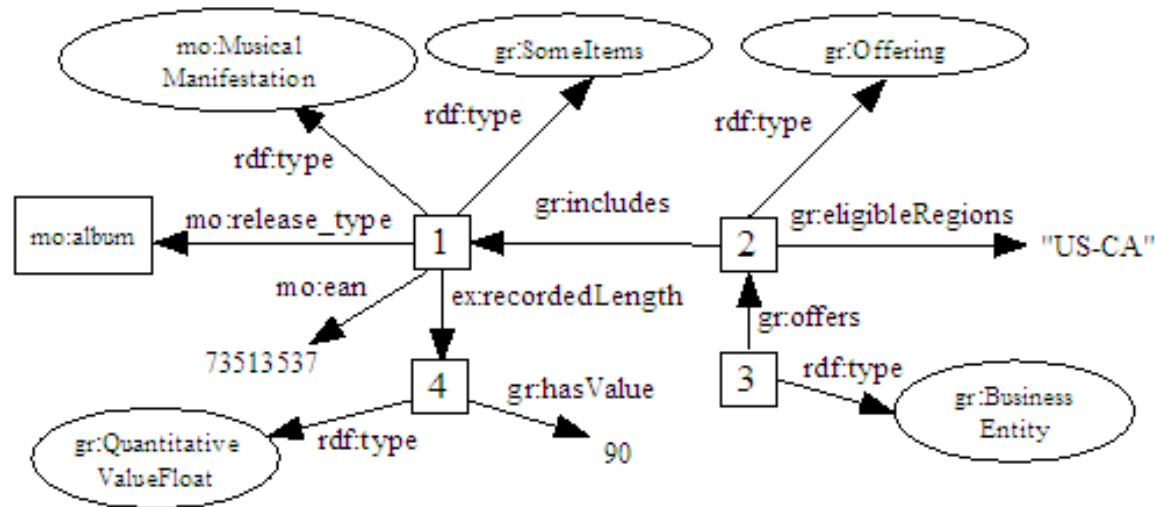
PURO: OBML pro relačně zaměřené ontologie

- PU: jednotliviny (Particulars)
vs. obecniny (Universals)
 - Neboli: **individua** (nelze jim přiřadit instance) vs. **typy**
- RO: **vztahy** (Relationships) vs. **objekty** (Objects)
- Proč právě tyto ontologické distinkce?
 - V principu intuitivní
 - Podobnost se základní syntaktickou „výbavou“ OWL:
individua vs. třídy
třídy vs. vlastnosti

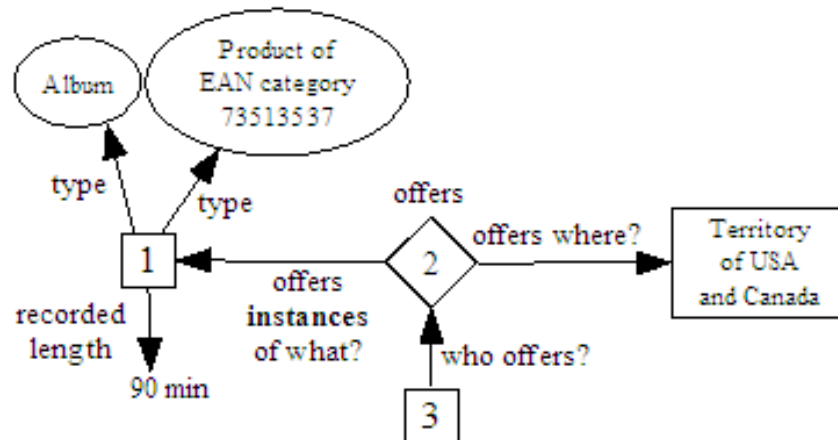
OFM vs. OBM: prodej hudebního alba

- Graf OFM (OWL)

- využívá slovníky Music Ontology a GoodRelations



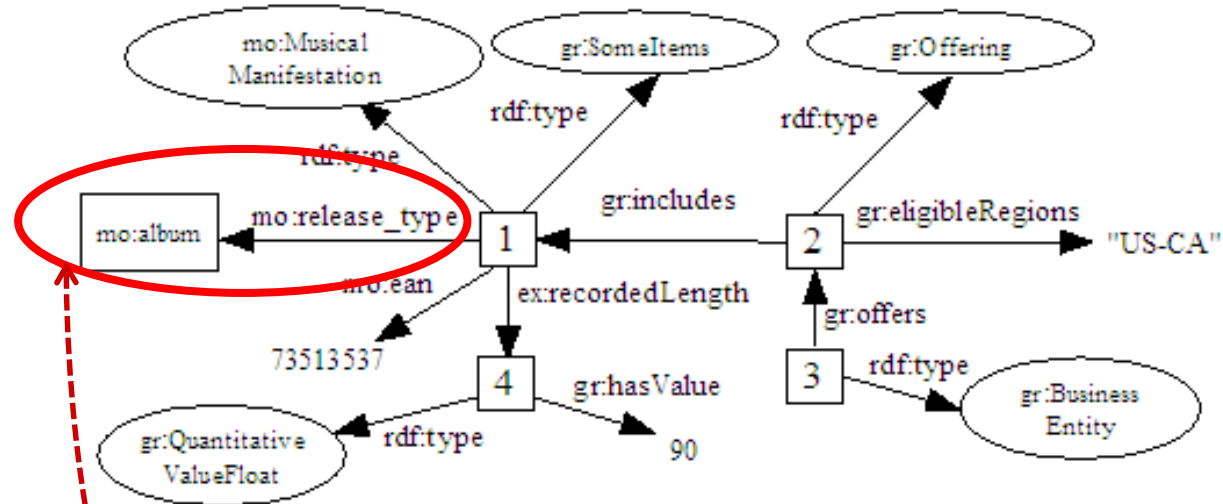
- Graf OBM (PURO)



OFM vs. OBM: prodej hudebního alba

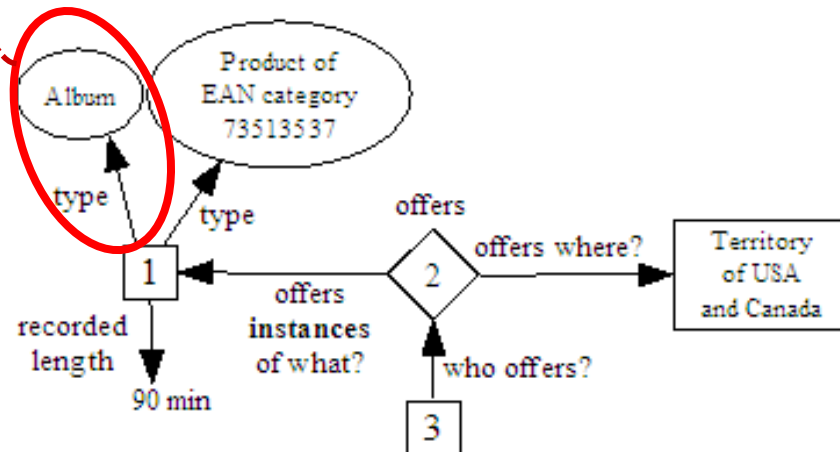
- Graf OFM (OWL)

- využívá slovníky Music Ontology a GoodRelations



*„Instanciace“
manifestovaná
jako objektová
vlastnost*

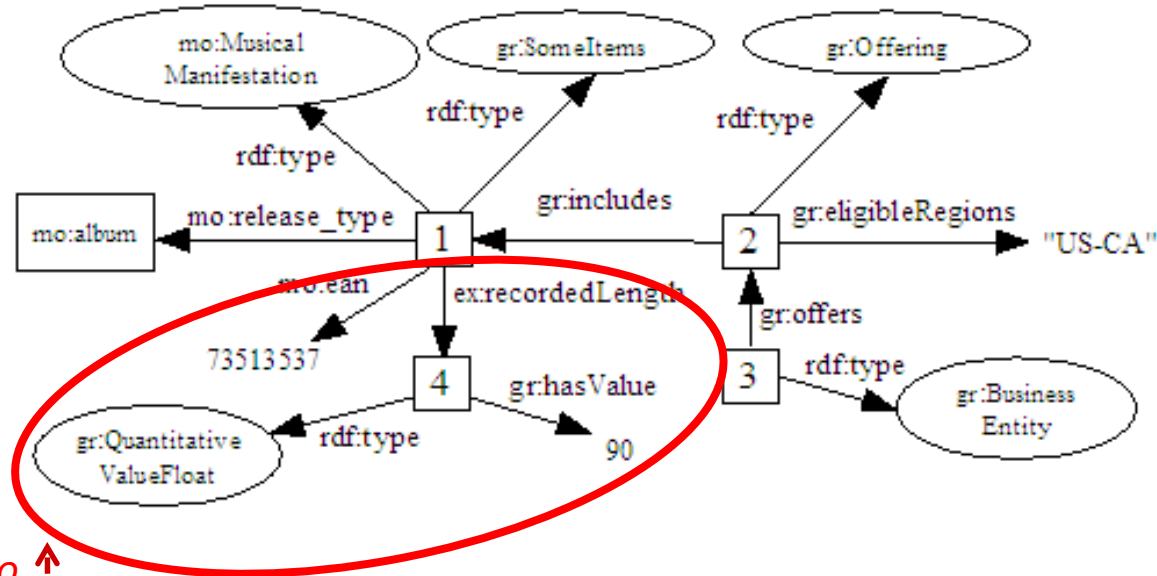
- Graf OBM (PURO)



OFM vs. OBM: prodej hudebního alba

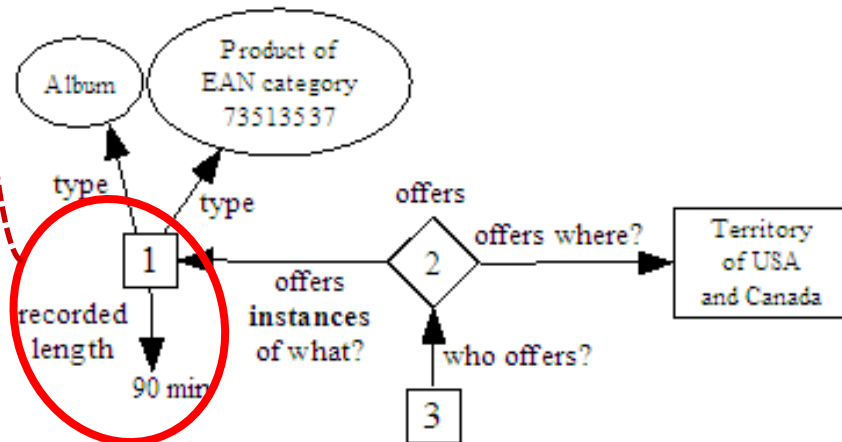
- Graf OFM (OWL)

- využívá slovníky Music Ontology a GoodRelations



„Value“ manifestovaná jako řetězec objektové a datové vlastnosti

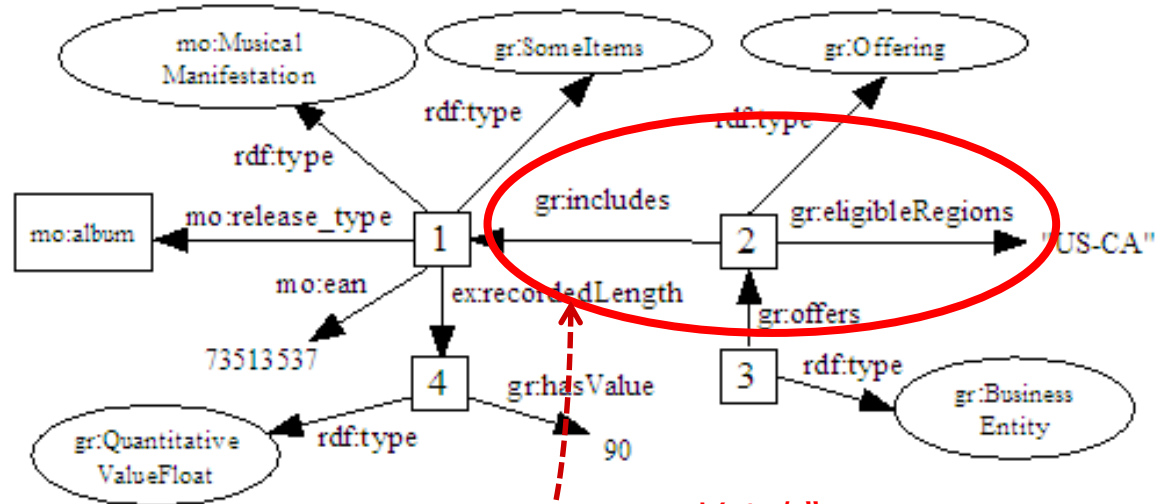
- Graf OBM (PURO)



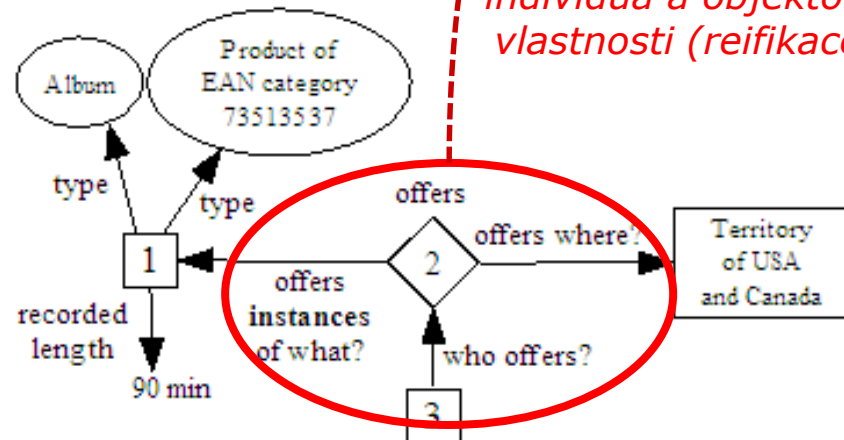
OFM vs. OBM: prodej hudebního alba

- Graf OFM (OWL)

- využívá slovníky Music Ontology a GoodRelations



- Graf OBM (PURO)

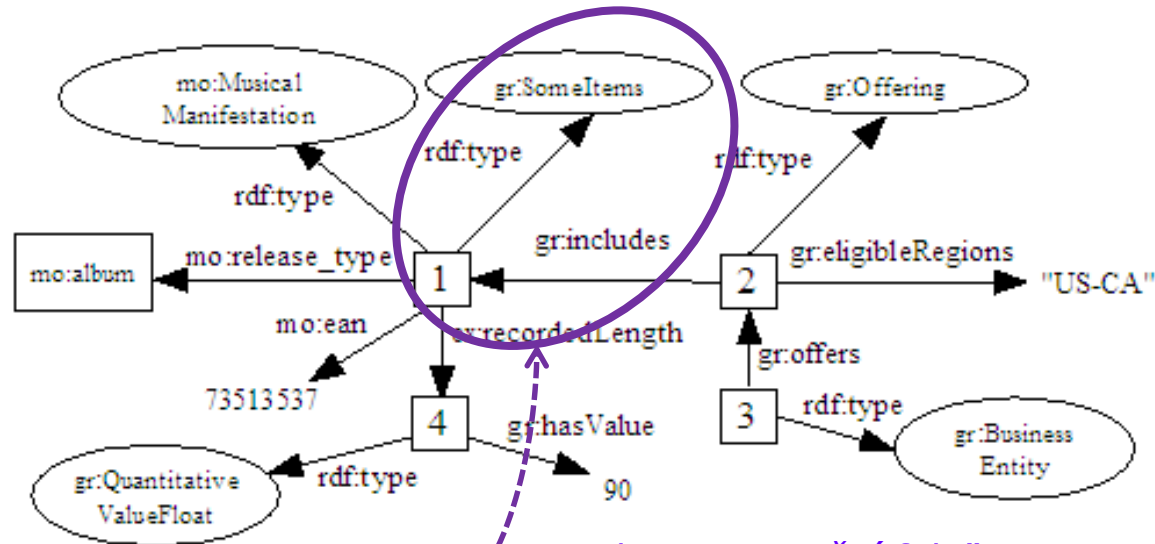


„Vztah“ manifestovaný jako individua a objektové vlastnosti (reifikace)

OFM vs. OBM: prodej hudebního alba

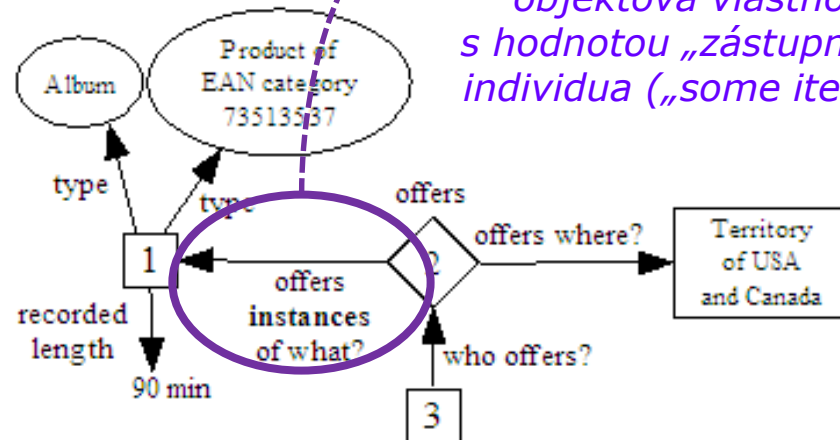
- Graf OFM (OWL)

- využívá slovníky Music Ontology a GoodRelations



„Multi-instantiační fakt manifestovaný jako objektová vlastnost s hodnotou „zástupného“ individua („some items“)

- Graf OBM (PURO)



Primitiva PURO OBML

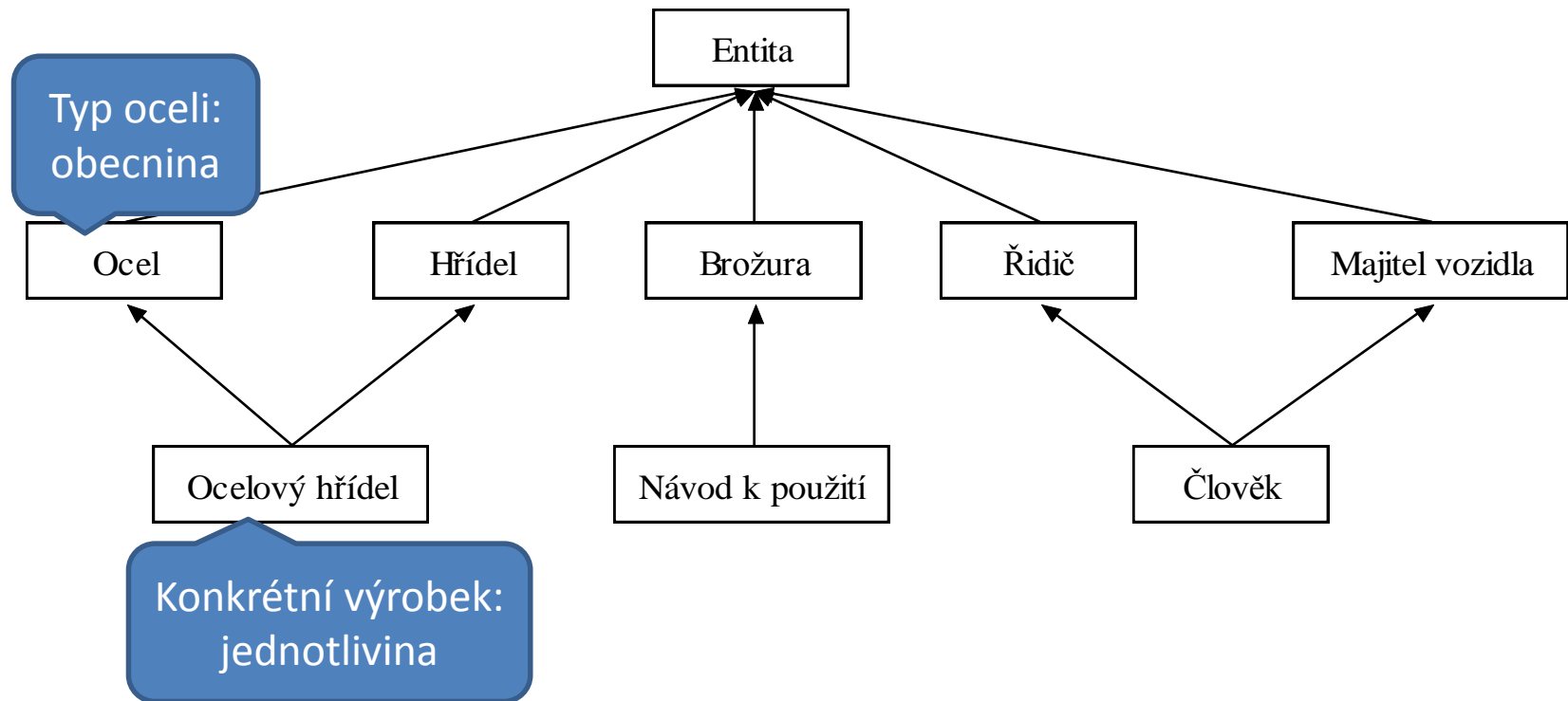
- Používá se prefix „*B*“, jako „background“
- *B*-objekt
- *B*-typ
 - *B*-typy 1. úrovně obsahují *B*-typy 2. úrovně
 - *B*-typy 2. úrovně obsahují *B*-objekty
 - ...
 - [ex:myCDno123](#) je *B*-objekt
 - [ex:CBS1992CD_YoYoMa_JSBach_SixCelloSuites](#) je *B*-typ 1. úrovně
 - [mo:Album](#) je *B*-typ 2. úrovně
 - [mo:ReleaseType](#) je *B*-typ 3. úrovně

Primitiva PURO OBML

- B -vztah („ B -relationship“)
 - mezi dvěma nebo více B -objekty nebo B -typy
- Jeho „univerzálním“ protějškem je B -relace („ B -relation“)
- B -valuace
 - přiřazení kvantitativní hodnoty
- Jejím „univerzálním“ protějškem je B -atribut

- B -vztahy jsou různých druhů
 - B -instanciace
 - B -axiom
 - Lze převést na operace nad množinami instancí
 - B -fact („všechny ostatní vztahy“)

Srovnání s OntoClean



Anotační technologie

- Nástroj **B-Annot**: plugin pro Protégé
- Umožňuje ručně anotovat ontologii (zpravidla datový slovník linked data) primitivy PURO, a anotaci následně uložit
 - **Generická** anotace odráží záměr tvůrců slovníku
 - Anotace **specifická pro dataset** odráží způsob použití slovníkových entit ve specifickém datasetu
 - pro podporu tohoto typu anotace lze zobrazit **datasetové souhrny**: časté řetězce tříd a vlastností v daném datasetu (s odlišením sledovaného slovníku)

B-Annot

0.1 (http://xmlns.com/foaf/0.1/) : [http://xmlns.com/foaf/spec/]

File Edit View Reasoner Tools Refactor Window Help

0.1 (http://xmlns.com/foaf/0.1/) Search for entity

Data Properties Individuals OWLViz DL Query B-Annotation

Active Ontology Entities Classes Object Properties

B-Annotation:

Create B-Annotation:

Generic Dataset-specific

Load/Save B-Annotation

Save... Load...

| Dataset URI | Total no. of triples | No. of triples usi... | Entities used | % of all triples | Call summi |
|-------------|----------------------|-----------------------|---------------|------------------|------------|
| eures | 7101481 | 6081 | 8 | 0,09 | |
| europena | 117118947 | 0 | 0 | 0 | |
| geonames | 114390681 | 6081 | 8 | 0,01 | |
| ookaboo | 51088319 | 7 | 1 | 0 | |
| worldbank | 87351631 | 14583 | 2 | 0,02 | |

Start annotating

0.1 (http://xmlns.com/foaf/0.1/) : [http://xmlns.com/foaf/spec/]

File Edit View Reasoner Tools Refactor Window Help

0.1 (http://xmlns.com/foaf/0.1/) Search for entity

Object Properties Data Properties Individuals OWLViz DL Query B-Annotation

Active Ontology Entities Classes

B-Annotation:

Create B-Annotation:

Generic Dataset-specific

Load/Save B-Annotation

Save... Load...

Dataset-specific B-Annotation - geonames

Entities found in dataset

http://www.w3.org/2000/01/rdf-schema#Class
http://www.w3.org/2002/07/owl#Thing
http://www.w3.org/2004/02/skos/core#Concept
http://xmlns.com/foaf/0.1/Agent

Selected entity from ontology: <http://xmlns.com/foaf/0.1/Agent> Choose a property:

| Entity Type | Entity Name | Entity Property | Remove |
|-------------|-----------------------------------|-----------------|--------|
| C | <http://xmlns.com/foaf/0.1/Agent> | CO | |

Object property hierarchy
Individuals
Data property hierarchy
Class hierarchy
Class hierarchy: Agent

- Thing
 - Agent ≡ Agent
 - Agent ≡ Agent
 - Class
 - Concept
 - Document
 - LabelProperty
 - OnlineAccount
 - Person
 - Project
 - SpatialThing

Možné využití anotací dle PURO

- Testování konceptuální koherence
 - jednotlivého datového slovníku / ontologie
 - mapování napříč slovníky
- Alternativní vizualizace dat
- Jednotící rámec pro související logicko-strukturní ontologické vzory
 - *Viz příspěvek na K-CAP'13 v seznamu literatury*
- Vhled do existujících slovníků, jejich utřídění, vzory pro návrh nových
- Extrakce „minimálních uzavřených popisů“ entit v RDF (CBD - „concise bounded descriptions“)

Shrnutí

- Ontologické inženýrství přichází na „divoký“ sémantický web až ve druhé vlně, nemůže proto aspirovat na roli „proaktivního“ stanovitele pravidel
- Může ale sehrát velmi užitečnou roli pro kvalitu a použitelnost existujících dat a slovníků
 - detekování a opravy chyb
 - nalézání pravidelností
 - srozumitelné přiblížení člověku
- Příklady „reaktivních“ projektů: PatOMat, ORE, PURO
- Příští... nebo přes příští generace sémantického webu bude opět výrazně „ontologická“!

Poděkování

- Spolupracovníci na projektech
 - **Ondřej Zamazal, Marek Dudáš, Ján Černý:**
návrh a implementace nástrojů PatOMat,
případové studie a experimenty
 - ***Lorenz Bühmann:***
spolupráce na integraci ORE a PatOMat
 - **Miroslav Vacura, Martin Homola, Ján Kluka:**
spolupráce na návrhu modelu PURO
 - **Simone Serra:**
implementace anotačního nástroje B-Annot

Vybraná literatura

- Šváb-Zamazal, Svátek, Iannone: *Pattern-Based Ontology Transformation Service Exploiting OPPL and OWL-API*. In: **EKAU'10**, Springer LNCS, 2010.
- Šváb-Zamazal, Dudáš, Svátek: *User-Friendly Pattern-Based Transformation of OWL Ontologies*. In: **EKAU'12**, Springer LNCS, 2012. Best Demo Award.
- Černý, Zamazal, Svátek: *Web-based Portal of Applicable Ontology Transformation Patterns*. In: **Znalosti'13** (sekce posterů), VŠB-TU, 2013.
- Dudáš, Svátek, Török, Zamazal, Rodriguez, Hepp: *Semi-automated Structural Adaptation of Advanced E-Commerce Ontologies*. In: **EC-Web'13**, Springer LNBIP, 2013.
- Lehmann, Bühmann: *ORE - A Tool for Repairing and Enriching Knowledge Bases*. In: **ISWC'10**, Springer LNCS, 2010.
- Zamazal, Bühmann, Svátek: *Checking and repairing ontological naming patterns using ORE and PatOMat*. In: **WoDOOM'13**, CEUR-WS, Vol.999, 2013.
- Guarino, Welty: *An Overview of OntoClean*. In: **The Handbook on Ontologies**, pp. 151–172, Springer-Verlag, 2009.
- Svátek, Homola, Kluka, Vacura: *Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models*. In: **OWLED'13**, CEUR-WS, vyjde v říjnu 2013.
- Svátek, Homola, Kluka, Vacura: *Mapping Structural Design Patterns in OWL to Ontological Background Models*. In: **K-CAP'13**, ACM, 2013.