Modularity in Ontologies: Introduction (Part A)

Thomas Schneider[1] *Dirk Walther*[2]

[1]Department of Computer Science, University of Bremen, Germany

[2]Center for Advancing Electronics, Technical University of Dresden, Germany

ESSLLI, 12 August 2013

Course Objective

- Ontologies are widely used as means to represent conceptualisations within a domain. In Computer Science, in particular, ontologies mainly provide a reference vocabulary.
- Many ontologies have been developed in several areas broad and diverse such as life sciences, health-care, linguistics, geosciences, etc.
- Examples: SNOMED CT, FMA, GALEN, GO, NCI, etc.
- We regard ontologies as logical theories.
- Challenge: Provide automatic support for sharing and reuse of (large) ontologies as well as their design and maintenance.
- Modularity is a key concept in tackling the challenges.
- We will define the notion of modularity for ontologies and provide an overview on its properties and uses as well as an introduction to related technical results.

(Tentative) Course Outline

- Monday: introduction (Thomas and Dirk)
- Tuesday: module extraction and its formal foundations (Thomas and Dirk)
- Wednesday: module extraction (Thomas and Dirk)
- Thursday: atomic decomposition (Thomas)
- Friday: recent advances/current work (Dirk)

Prerequisites:

- basic understanding of Description Logic *or* related logic formalisms
- basic knowledge of complexity theory

Knowledge Representation (KR)

General goal of KR: “develop formalisms for providing high-level descriptions of the world that can be effectively used to build intelligent applications” [Brachman and Nardi, 2003]

Requirements to a KR system:

- well-defined syntax and unambiguous semantics (machine processable)
- appropriate abstraction level (relevant vs. irrelevant aspects)
- reasoning about represented knowledge (esp. drawing of inferences of implicit from explicit knowledge)
- practical reasoning tools

Knowledge Representation (KR)

Early KR approaches:

- Semantic Networks [Quillian, 1967]
- Frame Systems [Minsky, 1981]

⇒ problem: no formal semantics

Logical formalisms:

- Logics have a formal syntax and semantics.
- Various logics with different expressivity are available.
- There are reasoning algorithms for many decidable logics.
- Optimised reasoning systems are available.

Logical Theories

- Logic = formal language L + formal semantics '$\models$'
- Logical theory: a set T of L-formulas (theorems) closed under logical consequences

$$\text{if } T \models \varphi, \text{ then } \varphi \in T$$

- Here we mostly consider finite axiomatisations of a theory and call these ontologies.

Applications of Theories

- propositional logic: constraint satisfaction problems, planning
- first-order logic: specification of graphs, datatypes
- temporal logic: specification of hard- and software systems
- epistemic logic: specification of agents' knowledge and belief
- non-monotonic logic: default reasoning, abductive reasoning, belief revision
- description logic: specification of vocabulary in an ontology (terminology)

Ontologies

In Computer/Information Science, ontologies are “a formal, explicit specification of a shared conceptualisation” [Gruber, 1993]

An ontology:

- covers a domain of interest, e.g., medicine, biology, geography, linguistics, ESSLLI, etc.
- presents a model of the domain
- provides a vocabulary with names for objects, relations and classes and defines relationships between them
- can define names for terms from existing ones
- is presented in a formal language with formal semantics
- is shared among users

Ontologies

Examples of vocabulary classifications (taxonomies):

- astronomy: Secchi classes to classify stars (Secchi, 1877)
- biology: Linnaean taxonomy (Linnaeus, 1735) classifying organisms
- gastronomy: Bordeaux Wine Official Classification of 1855
- libraries: Dewey Decimal Classification (Dewey, 1876)
- medicine: International Statistical Classification of Diseases and Related Health Problems (ICD)
- common sense knowledge: OpenCyc (`opencyc.org`)
- websites: Open Directory Project (`dmoz.org`)

Ontologies

Repositories:

- NCBO BioPortal
  `http://bioportal.bioontology.org`
- Oxford Ontology Repository
  `http://www.cs.ox.ac.uk/isg/ontologies/lib/`
- TONES Ontology Repository
  `http://owl.cs.manchester.ac.uk/repository/`

Ontologies

Examples of ontologies in medicine and biology:

- Gene Ontology (GO)
  `geneontology.org`
  - provides vocabularies for the annotation of gene products
  - `protein_tag` $\sqsubseteq$ `molecular_function`
- National Cancer Institutes Thesaurus (NCI)
  `ncicb.nci.nih.gov`
  - `HIV_Budding` $\sqsubseteq$ `Virus-Cell_Membrane_Interaction`

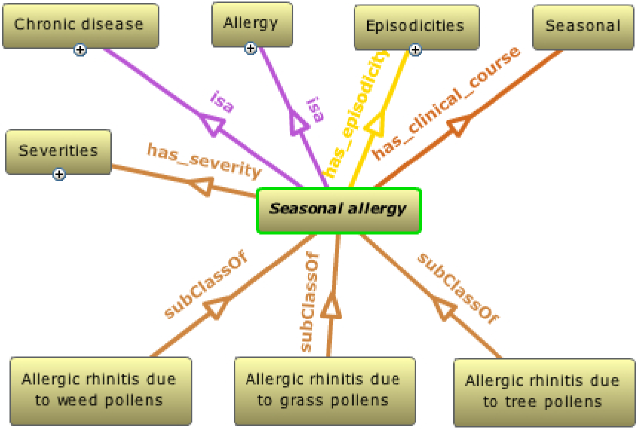
SNOMED CT

Systematized Nomenclature of Medicine Clinical Terms
`ihtsdo.org`

- provides medical terminology
- example:

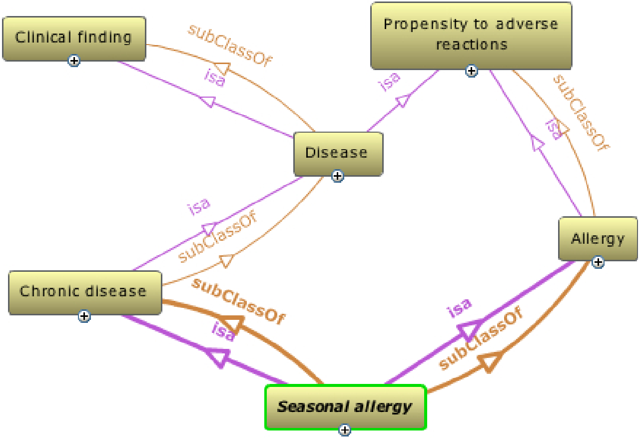
$$\begin{aligned}\texttt{Seasonal\_Allergy} \sqsubseteq\ & \texttt{Chronic\_disease} \sqcap \texttt{Allergy} \\ & \sqcap \exists \texttt{has\_severity.Severities} \\ & \sqcap \exists \texttt{has\_episodicity.Episodicities} \\ & \sqcap \exists \texttt{has\_clinical\_course.Seasonal}\end{aligned}$$

SNOMED CT



- neighbourhood of `Seasonal_Allergy` (using `bioportal.bioontology.org`)

SNOMED CT



- class hierarchy (taxonomy) from `Seasonal_Allergy` (using `bioportal.bioontology.org`)

Semantic Web

- “provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries” [W3C, 2010]
- extends World Wide Web with meta data (e.g. annotations) about the pages and how they relate to each other
- goal: machines automatically process information on the web (find, share, combine, act upon, reason with information, etc.) $\Rightarrow$ “intelligent machines”
- name coined by Tim Berners-Lee
- some functionality:
  - answer queries involving background knowledge
  - access information in data repositories
  - use web services
  - delegate tasks to agents

Examples of Ontology languages

- formalism for knowledge representation
  - ER- and UML diagrams
  - Conceptual Graphs
  - Datalog and rule-based languages
  - DLs and higher-order logics
- traditional ontology specification languages
  - Ontolingua
  - Operational Conceptual Modeling Language (OCML)
  - Frame Logic
- web standards and W3C recommendations
  - eXtended Markup Language (XML)
  - Resource Description Framework (RDF) and RDF Schema
  - Web Ontology Language (OWL)

Web Ontology Language (OWL)

- ontology language for the Semantic Web with formally defined meaning
- designed to facilitate ontology development and sharing via the Web
- provide classes, properties, individuals, and data values
- a standard for ontologies in applications in the web (also used independently of the web)
- RDF/XML-based syntax
- W3C standards: OWL (2004), OWL 2 (2009) (technical reports available under `www.w3.org/TR/`)
- profiles (sub-languages) to trade expressive power for performance guarantees of reasoning

Description Logic

- "family of logic-based knowledge representation formalisms"
  ⇒ fulfill requirements to a KR system
- W3C recommends to base OWL languages onto DL
- expressivity vs. computational complexity
- DLs define classes, properties/relations and objects using concepts, roles and individuals
- concept language of DLs:
  - *concept names* are names for groups of objects
  - *role names* are names for relations between objects
  - *individual names* are names for objects
  - *constructors* relate names for concepts, roles and individuals

Example: a terminology of ESSLLI

- classes (concepts): Person, Course, Lecturer, Attendant, ...
- relations (roles): attends, gives, likes, ...
- objects (individuals): Thomas, x, y, ...
- definitions:
  - $\mathtt{Lecturer} \equiv \mathtt{Person} \sqcap \exists \mathtt{gives.Course}$
  - $\mathtt{Attendant} \equiv \mathtt{Person} \sqcap \exists \mathtt{attends.Course}$
  - $\mathtt{Registrant} \equiv \mathtt{Person} \sqcap \mathtt{Registered}$
- assertions:
  - $\mathtt{Lecturer}(\mathtt{Thomas})$, $\mathtt{Attendant}(\mathtt{x})$, $\mathtt{Attendant}(\mathtt{y})$
  - $\mathtt{gives}(\mathtt{Thomas}, \mathtt{mod\text{-}course})$, $\mathtt{likes}(\mathtt{x}, \mathtt{mod\text{-}course})$, $\mathtt{likes}(\mathtt{x}, \mathtt{y})$
- constraints:
  - $\mathtt{Workshop} \sqsubseteq \forall \mathtt{attended\_by.Registrant}$
  - $\mathtt{attended\_by} \equiv \mathtt{attends}^{-1}$

DLs and FOL

- DLs can be embedded into FOL
  - concepts correspond to unary predicates
  - roles correspond to binary predicates
  - no more than 2 variables under the scope of a quantifier (exception: transitive roles, number restrictions, etc.)
  - individuals correspond to constants
  - no function symbols
- DLs are usually decidable

The basic Description Logic ALC

- signature: countably infinite supply of concept names $A, B, \ldots$, role names $r, s, \ldots$ and individual names $a, b, \ldots$
- syntax:

$$C, D ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.D$$

- individual assertions:
  - $C(a)$
  - $r(a, b)$
- axioms:
  - $C \sqsubseteq D$
  - $C \equiv D$
- ABox: finite set of individual assertions
- TBox: finite set of axioms

ALC Interpretations

- interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
  - $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for all concept names A
  - $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for all role names r
  - $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ for all individual names a

Name	Syntax	Semantics
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}}\}$
universal restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in C^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}}\}$

ALC Semantics

An interpretation $\mathcal{I}$ satisfies:

- concept inclusion: $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- concept equation: $C \equiv D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$
- TBox: T iff $\mathcal{I}$ satisfies all axioms in T ($\mathcal{I}$ is a model of T)

- concept assertion: $C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- role assertion: $r(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
- ABox: A iff $\mathcal{I}$ satisfies all assertions in A ($\mathcal{I}$ is a model of A)

Common Reasoning Tasks

(1) Subsumption of concepts C, D wrt. TBox T
  - Does $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ hold in all models of T?

(2) Satisfiability of concept C wrt. TBox T
  - Is there a model $\mathcal{I}$ of T such that $C^{\mathcal{I}} \neq \emptyset$?

(3) Consistency of KB $K = (T, A)$
  - Is there a common model of T and A?

(4) Instance checking of individual a in concept C wrt. KB $K = (T, A)$
  - Does $a^{\mathcal{I}} \in C^{\mathcal{I}}$ hold in all models $\mathcal{I}$ of K?

(5) Query answering
  - Given a KB $K = (T, A)$, a query $q(\vec{x})$ and a tuple $\vec{a}$ of individual names from A, does $\mathcal{I}$ *satisfy* $q(\vec{a})$ for all models $\mathcal{I}$ of K?

Light-weight DLs

- provide tractable reasoning
- DL-Lite family [Calvanese et al., 2007]
  - conceptual modelling (capture much of ER- and UML-diagrams)
  - designed to access large amounts of data via high-level conceptual interface (data integration, querying instance data using background theories)
- EL family [Baader, Brandt, Lutz, 2005]
  - captures large biomedical ontologies like SNOMED CT, NCI thesaurus
- common restrictions: no disjunction, no universal restriction

Expressive DL SROIQ

- OWL 2
- provides the ontology developer with any desirable (but reasonable) expressive means for easy and intuitive modelling
- reasoning is 2NExpTime-complete [Kazakov, 2008]
- $\mathcal{ALC}$ extended with:
  - nominals
  - qualified number restrictions
  - conditions on roles: (ir)reflexivity, symmetry, transitivity and universality
  - conditions between roles: complex role inclusions and disjointness

Some DL-Reasoners

- EL+:
  - CEL (http://lat.inf.tu-dresden.de/systems/cel)
- SHIQ:
  - KAON2 (http://kaon2.semanticweb.org)
- SROIQ:
  - FaCT++ (http://owl.man.ac.uk/factplusplus/)
  - HermiT (http://hermit-reasoner.com)
  - Pellet (http://clarkparsia.com/pellet/)
  - RacerPro (http://racer-systems.com)
- ...

ORE 2013 - The OWL Reasoner Evaluation Workshop

- 2nd edition of OWL reasoner performance competition
  `http://ore2013.cs.manchester.ac.uk`
- 14 reasoners: TrOWL, Konclude, TReasoner, HermiT, MORe, FaCT++, Jfact, Chainsaw, WSClassifier, ELK, jcel, SnoRocket, ELepHant, BaseVISor
- input ontologies:
  - ontology repositories
  - user submitted hard ontologies
- reasoning tasks: consistency, classification, satisfiability

More tool support?

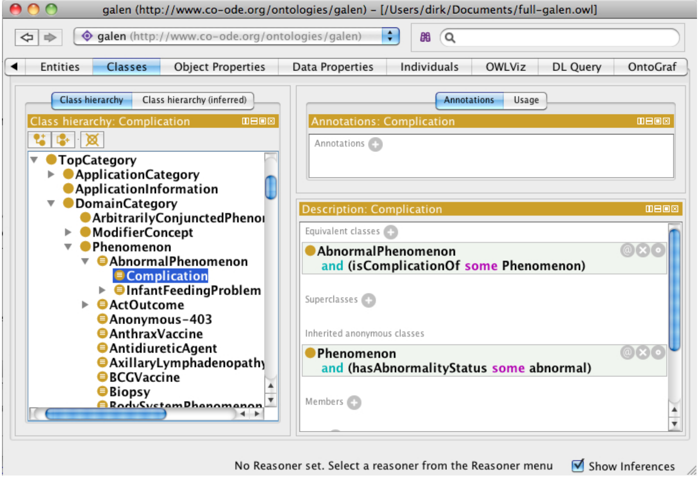
- development of ontologies
- editing an OWL ontology with RDF/XML syntax (`full-galen.owl` in a text editor)

full-galen.owl

```
<?xml version="1.0"?>^M
<rdf:RDF^M
    xmlns:rss="http://purl.org/rss/1.0/"^M
    xmlns="http://www.co-ode.org/ontologies/galen#"^M
    xmlns:jms="http://jena.hpl.hp.com/2003/08/jms#"^M
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"^M
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"^M
    xmlns:owl="http://www.w3.org/2002/07/owl#"^M
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"^M
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"^M
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.co-ode.org/ontologies/galen" >^M
  <rdf:Description rdf:nodeID="A0">^M
    <owl:intersectionOf rdf:nodeID="A1"/>^M
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>^M
  </rdf:Description>^M
  <rdf:Description rdf:about="#CerebroSpinalFluid">^M
    <rdfs:subClassOf rdf:nodeID="A2"/>^M
    <rdfs:subClassOf rdf:resource="#NAMEDBodySubstance"/>^M
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>^M
    <rdfs:subClassOf rdf:nodeID="A3"/>^M
  </rdf:Description>^M
  <rdf:Description rdf:nodeID="A4">^M
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>^M
    <owl:onProperty rdf:resource="#hasStructuralComponent"/>^M
    <owl:someValuesFrom rdf:resource="#RadialRecurrentArtery"/>^M
```

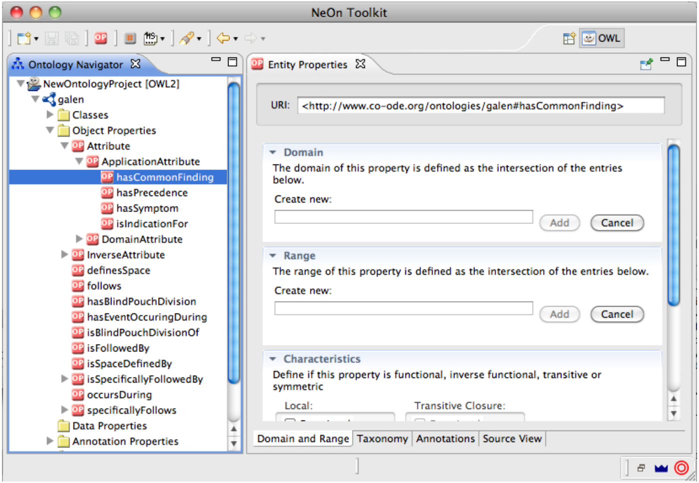
Ontology Editor

- full-galen.owl in Protégé
  http://protege.stanford.edu



Ontology Editor

- `full-galen.owl` in NeOn Toolkit
  `http://neon-toolkit.org`



And now ...

Part B (Thomas): overview on modularity in ontologies