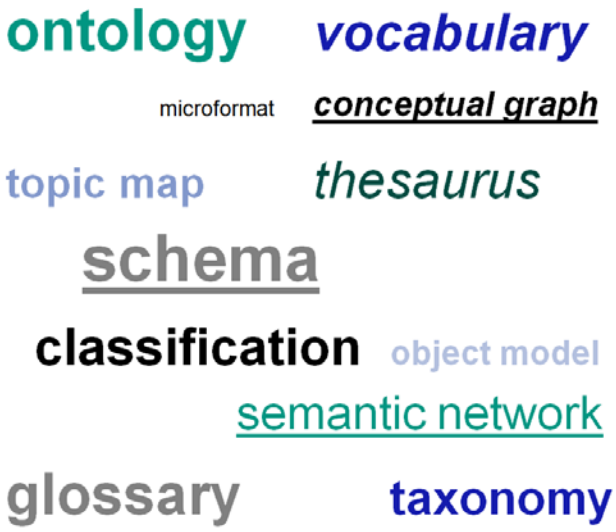
Ontology design

Elena Simperl & Denny Vrandečić, AIFB, Karlsruhe Institute of Technology, DE

With contributions from "Linked Data: Survey of Adoption", Tutorial at the 3rd Asian Semantic Web School ASWS 2011, Incheon, South Korea, July 2011 by Aidan Hogan, DERI, IE

Ontologies in Computer Science

- An ontology defines
  - Concepts
  - Relationships
  - Any other distinctions relevant to capture and model knowledge from a domain of interest
- Ontologies are used to
  - Share a common understanding about a domain among people or machines
  - Enable reuse of domain knowledge
- This is achieved by
  - Agree on meaning and representation of domain knowledge
  - Make domain assumptions explicit.
  - Separate domain knowledge from the operational knowledge
- Application areas
  - Natural language processing
  - Multimedia analysis
  - Machine learning
  - Digital libraries
  - Software engineering
  - Database design
  - …



Are ontologies just UML?

- Ontologies vs ER schemas
  - Semantic Web ontologies represented in Web-compatible languages, use Web technologies
  - They represent a shared view over a domain
- Ontologies vs UML diagrams
  - Formal semantics of ontology languages defined, languages with feasible computational complexity available
- Ontologies vs thesauri
  - Formal semantics, domain-specific relationships
- Ontologies vs taxonomies
  - Richer property types, formal semantics of the is-a relationship

Ontologies and Linked Data

- **Global, distributed dataspace build on a simple set of standards**
  - RDF, URIs, HTTP
- **Entities are connected by links**
  - creating a global data graph that spans data sources and
  - enables the discovery of new data sources
- **Provides for data-coexistence**
  - Everyone can publish data to the Web of Linked Data
  - Everyone can express their personal view on things
  - Everybody can use the vocabularies/schema that they like

Tasks:

1. **Make data available as RDF via HTTP**
2. **Set RDF links pointing at other data s**
3. **Make your data self-descriptive**

- **Aspects of self-descriptiveness**
  1. Enable clients to retrieve the schema
  2. Reuse terms from common vocabularies
  3. Publish schema mappings for proprietary terms
  4. Provide provenance metadata
  5. Provide licensing metadata
  6. Provide data-set-level metadata using voiD
  7. Refer to additional access methods using voiD

Content due to Chris Bizer

Ontologies and Linked Data

- Model pre-defined through the (semi-) structure of the data to be published
- Emphasis on alignment, especially at the instance level
- Stronger commitment to reuse instead of development from scratch

- Human vs machine-oriented consumption (using specific technologies)
- Trade-off between acceptance/ease-of-use and expressivity/usefulness

- Publication according to Linked Data principles

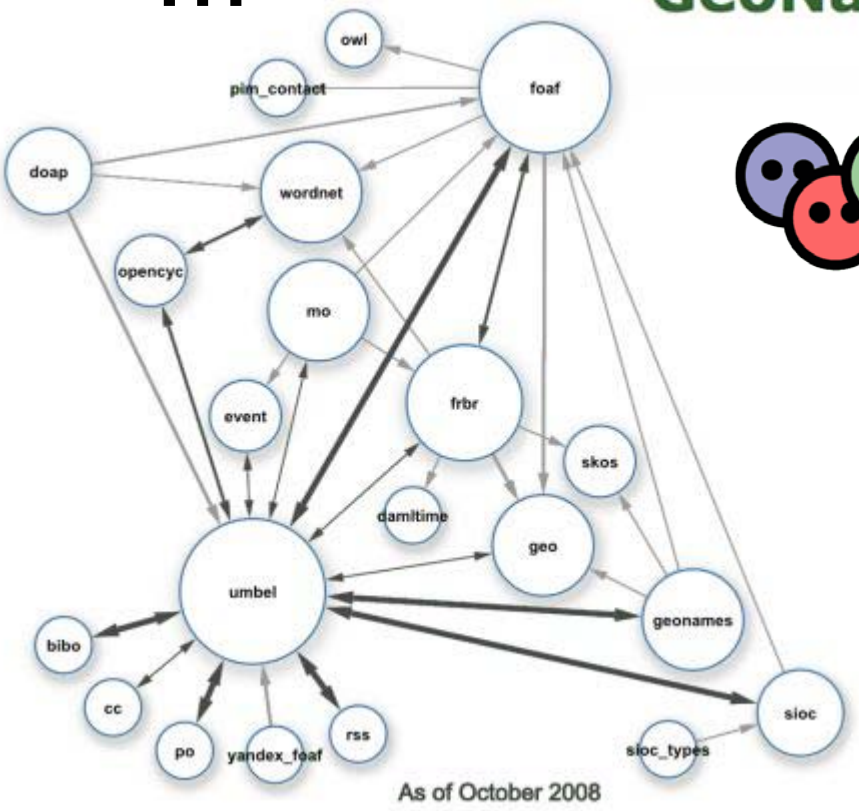
Reuse Terms from Common Vocabularies

- Common Vocabularies
  - **Friend-of-a-Friend** for describing people and their social network
  - **SIOC** for describing forums and blogs
  - **SKOS** for representing topic taxonomies
  - **Organization Ontology** for describing the structure of organizations
  - **GoodRelations** provides terms for describing products and business entities
  - **Music Ontology** for describing artists, albums, and performances
  - **Review Vocabulary** provides terms for representing reviews
- Common sources of identifiers (URIs) for real world objects
  - **LinkedGeoData** and **Geonames** locations
  - **GeneID** and **UniProt** life science identifiers
  - **DBpedia** wide range of things

Content due to Chris Bizer

(Linked) vocabularies overview

...



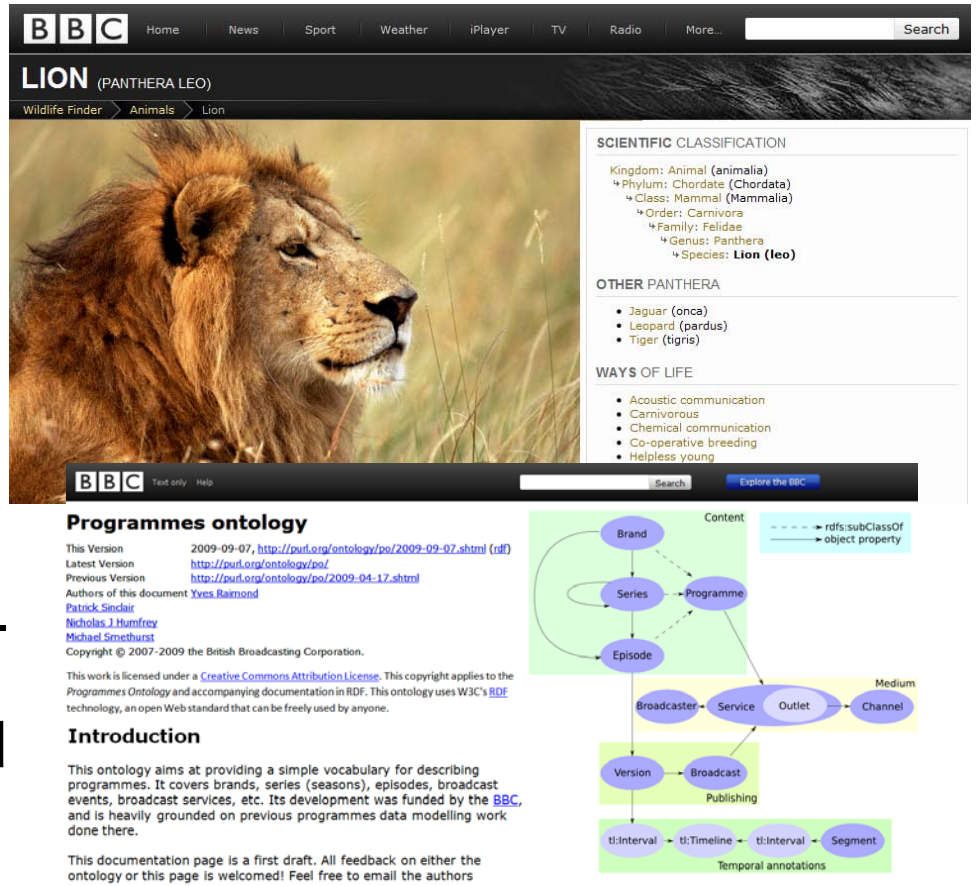
...

Image from http://blog.dbtune.org/public/.081005_lod_constellation_m.jpg, Giasson, Bergman

„Design for a world where Google is your homepage, Wikipedia is your CMS, and humans, software developers and machines are your users“

Example: BBC

- Various micro-sites built and maintained manually.
- No integration across sites in terms of content and metadata.
- Use cases
  - Find and explore content on specific (and related) topics.
  - Maintain and re-organize sites.
  - Leverage external resources.
- Ontology: One page per thing, reusing DBpedia and MusicBrainz IDs, different labels…



http://www.slideshare.net/reduxd/beyond-the-polar-bear

ONTOLOGY DESIGN LAB

Assignment 1

- Describe the automotive domain using 10-20 entities, attributes and relationships

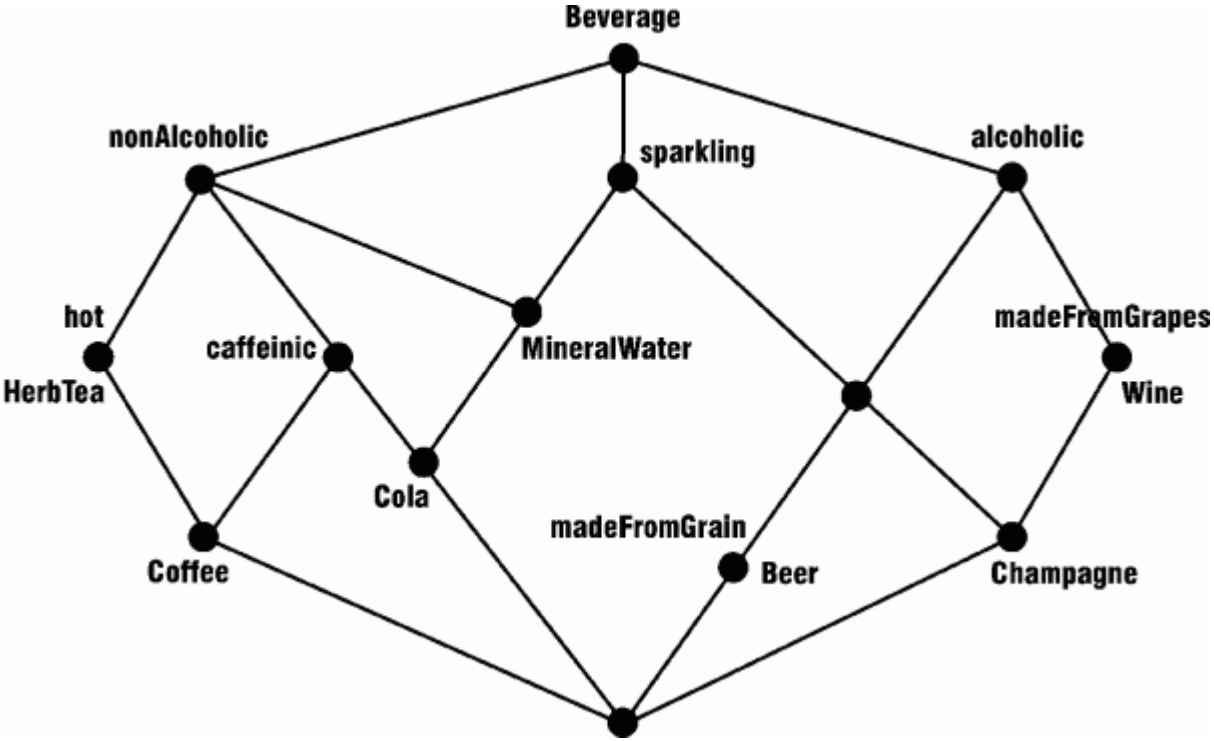
Assignment 2

- Imagine an online movie recommendation portal such as IMDB or GetGlue
- Develop an ontology for this domain
- Implement the ontology using an editor of your choice

Assignment 3

- What is the cardinality and existence of each of the following relationships in just the direction given? State any assumptions you have to make
  1. Husband to wife
  2. Student to degree
  3. Child to parent
  4. Player to team
  5. Student to course

Assignment 4



From
http://www.jfsowa.com/ontology/

Assignment 5

- Model the following statements
  - Barack Hussein Obama is the nominee of the Democratic Party for the office of President of the United States in the 2008 general election
  - Peter saw Van Gogh’s sunflowers in an MOMA exhibition at the Louvre in December last year

WHAT ONTOLOGIES ARE OUT THERE?

Life sciences and healthcare

The Open Biological and Biomedical Ontologies

Ontologies | Resources | Participate | About

The OBO Foundry is a collaborative experiment involving developers of science-based ontologies who are establishing a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain. The groups developing ontologies who have expressed an interest in this goal are listed below, followed by other relevant efforts in this domain.

In addition to a listing of OBO ontologies, this site also provides a statement of the OBO Foundry principles, discussion fora, technical infrastructure, and other services to facilitate ontology development. We welcome feedback and encourage participation.

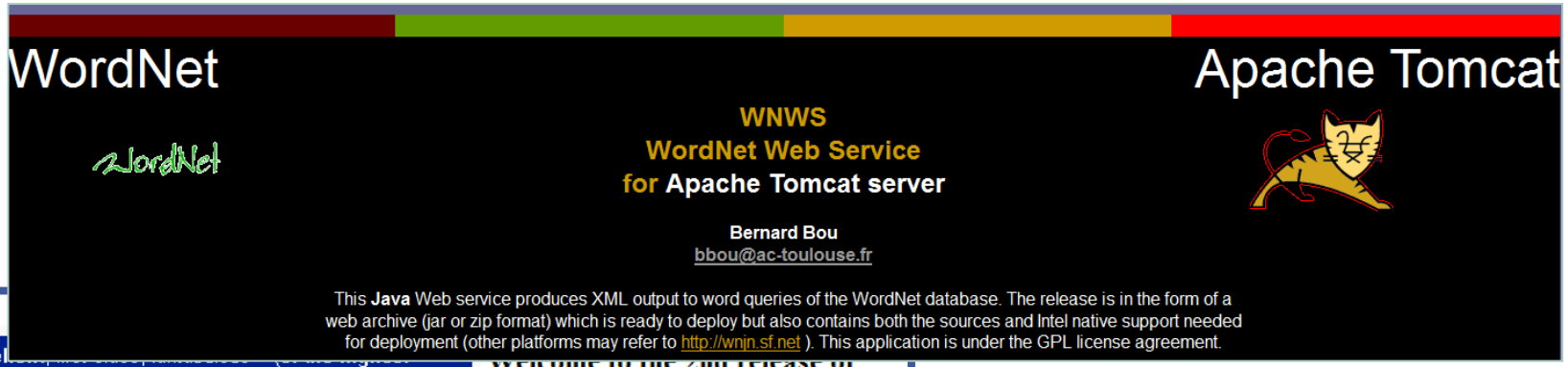
Click any column header to sort the table by that column. The s link to the term request trackers for the listed ontologies.

OBO Foundry ontologies

Title	Domain	Prefix	File	Last changed
Biological process	biological process	GO	gene_ontology_edit.obo	2010/07/26
Cellular component	anatomy	GO	gene_ontology_edit.obo	2010/07/26
Chemical entities of biological interest	biochemistry	CHEBI	chebi.obo	2010/07/07
Molecular function	biological function	GO	gene_ontology_edit.obo	2010/07/26
Phenotypic quality	phenotype	PATO	quality.obo	2010/07/25
PRotein Ontology (PRO)	proteins	PRO	pro.obo	2010/07/24
Xenopus anatomy and development	anatomy	XAO	xenopus_anatomy.obo	2009/12/02
Zebrafish anatomy and development	anatomy	ZFA	zebrafish_anatomy.obo	2010/06/14

WordNet

http://www.w3.org/TR/wordnet-rdf/



1. excel… **quality**; "made an excellent speech"; "the school has excellent teachers"; "a first-class mind")

```
(TOP (S (NP (JJ excellent) )
        (VP (VBZ is)
          (NP (NP (NN something) )
            (PP (IN of)
              (NP (DT the) (JJS highest) (NN quality) ) ) ) )
        (. .) ) )
```

excellent:JJ(x1) -> **of**:IN(x1, x2) **highest**:JJ(x1)
quality:NN(x1)

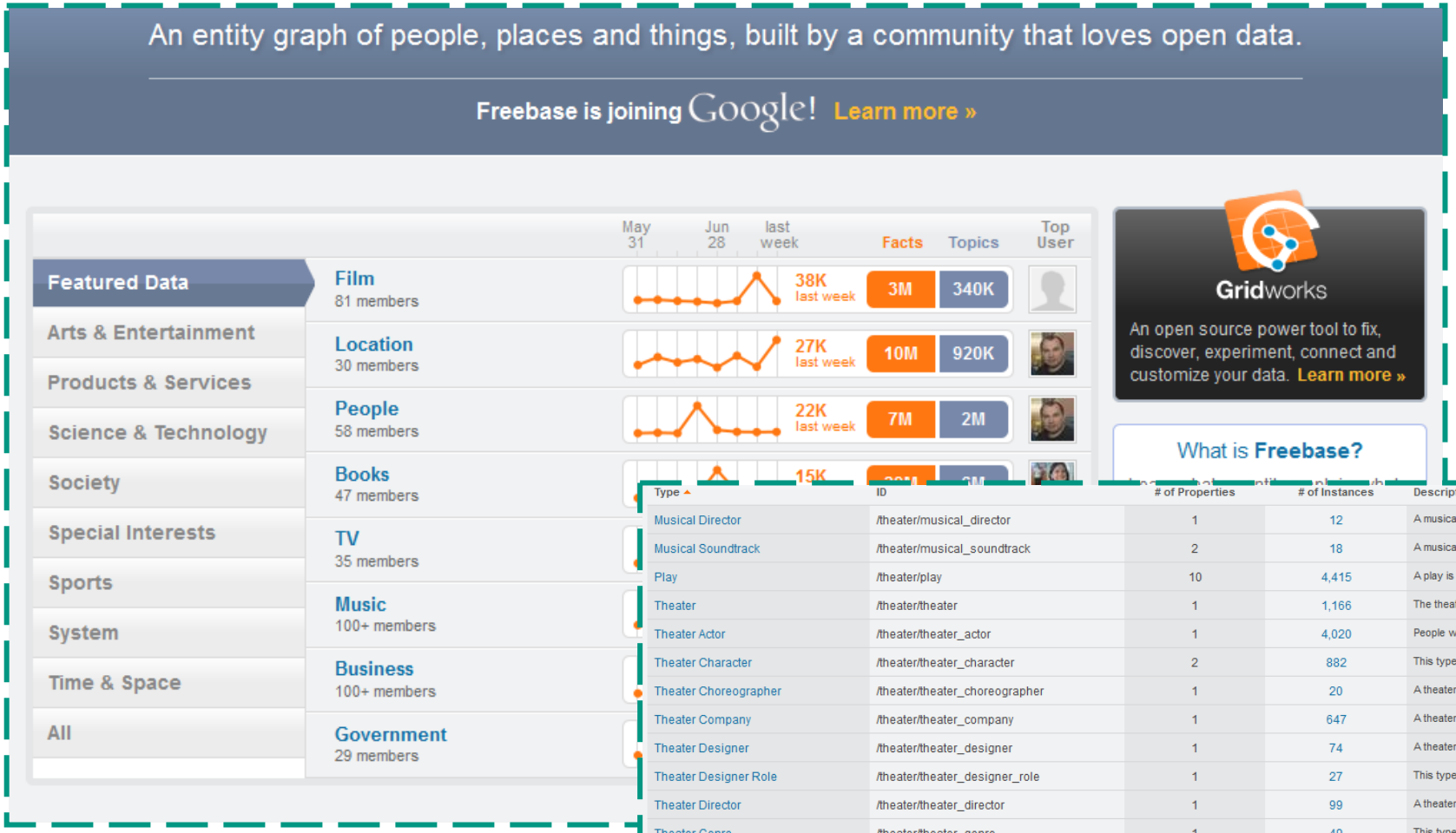
```
<wf pos="IN" >of</wf>
<wf pos="DT" >the</wf>
<wf pos="JJS" lemma="high" quality="silver" wnsn="1"
```

eXtended WordNet

The goal of this project is to develop a tool that takes as input the current or future versions of WordNet and automatically generates an eXtended WordNet that provides several important enhancements intended to remedy the present limitations of Word…

In the e… glosses a… transforr… words a…

Freebase



Type	ID	# of Properties	# of Instances	Description
Musical Director	/theater/musical_director	1	12	A musical director is a person w
Musical Soundtrack	/theater/musical_soundtrack	2	18	A musical soundtrack is a record
Play	/theater/play	10	4,415	A play is a work written to be pe
Theater	/theater/theater	1	1,166	The theater type is for performan
Theater Actor	/theater/theater_actor	1	4,020	People who have performed in a
Theater Character	/theater/theater_character	2	882	This type is for all characters tha
Theater Choreographer	/theater/theater_choreographer	1	20	A theater choreographer is some
Theater Company	/theater/theater_company	1	647	A theater company is a group of
Theater Designer	/theater/theater_designer	1	74	A theater designer is someone w
Theater Designer Role	/theater/theater_designer_role	1	27	This type represents possible po
Theater Director	/theater/theater_director	1	99	A theater director is someone wh
Theater Genre	/theater/theater_genre	1	49	This type is for all genres of play
Theater Producer	/theater/theater_producer	1	39	A theater producer is a person w
Theater Production	/theater/theater_production	15	599	A theater production is a product
Theater Production Staff Role	/theater/theater_production_staff_role	1	18	This type represents all positions
Theater Production Staffmember	/theater/theater_production_staff	1	33	This type is for any person who
Theatrical Composer	/theater/theatrical_composer	1	140	A theatrical composer is someon
Theatrical Lyricist	/theater/theatrical_lyricist	1	144	A theatrical lyricist is someone w

Dublin Core

Properties in the */terms/* namespace	abstract, accessRights, accrualMethod, accrualPeriodicity, accrualPolicy, alternative, audience, available, bibliographicCitation, conformsTo, contributor, coverage, created, creator, date, dateAccepted, dateCopyrighted, dateSubmitted, description, educationLevel, extent, format, hasFormat, hasPart, hasVersion, identifier, instructionalMethod, isFormatOf, isPartOf, isReferencedBy, isReplacedBy, isRequiredBy, issued, isVersionOf, language, license, mediator, medium, modified, provenance, publisher, references, relation, replaces, requires, rights, rightsHolder, source, spatial, subject, tableOfContents, temporal, title, type, valid
Properties in the legacy */elements/1.1/* namespace	contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, type
Vocabulary Encoding Schemes	DCMIType, DDC, IMT, LCC, LCSH, MESH, NLM, TGN, UDC
Syntax Encoding Schemes	Box, ISO3166, ISO639-2, ISO639-3, Period, Point, RFC1766, RFC3066, RFC4646, RFC5646, URI, W3CDTF
Classes	Agent, AgentClass, BibliographicResource, FileFormat, Frequency, Jurisdiction, LicenseDocument, LinguisticSystem, Location, LocationPeriodOrJurisdiction, MediaType, MediaTypeOrExtent, MethodOfAccrual, MethodOfInstruction, PeriodOfTime, PhysicalMedium, PhysicalResource, Policy, ProvenanceStatement, RightsStatement, SizeOrDuration, Standard

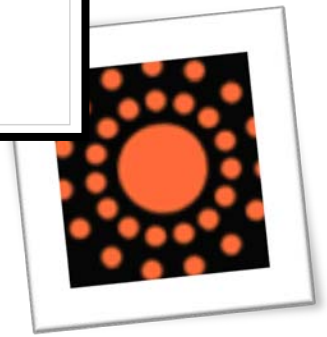


Table from http://dublincore.org/documents/dcmi-terms/

Friend Of A Friend

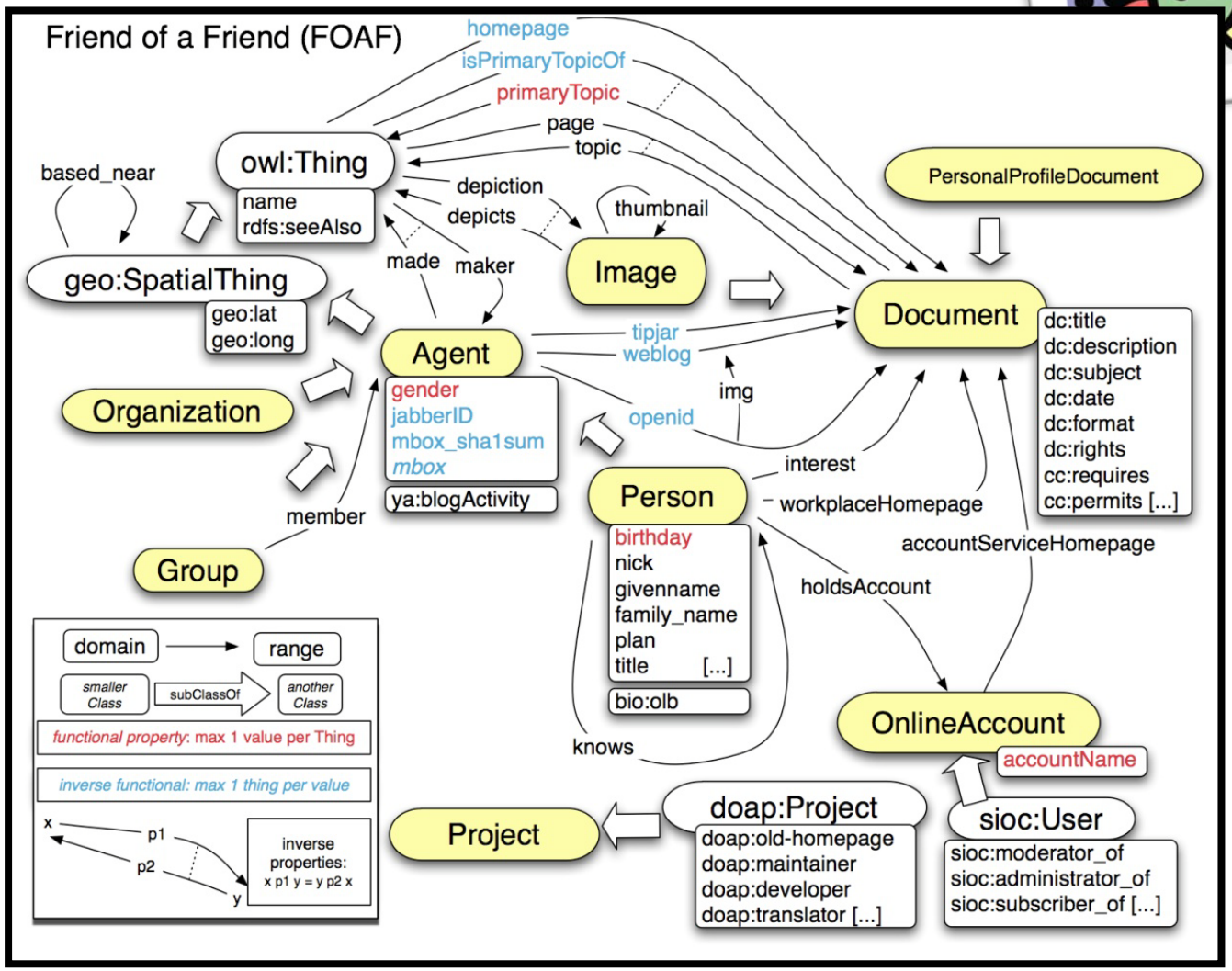


Image from http://www.deri.ie/fileadmin/images/blog/ : Breslin

Semantically Interlinked Online Communities

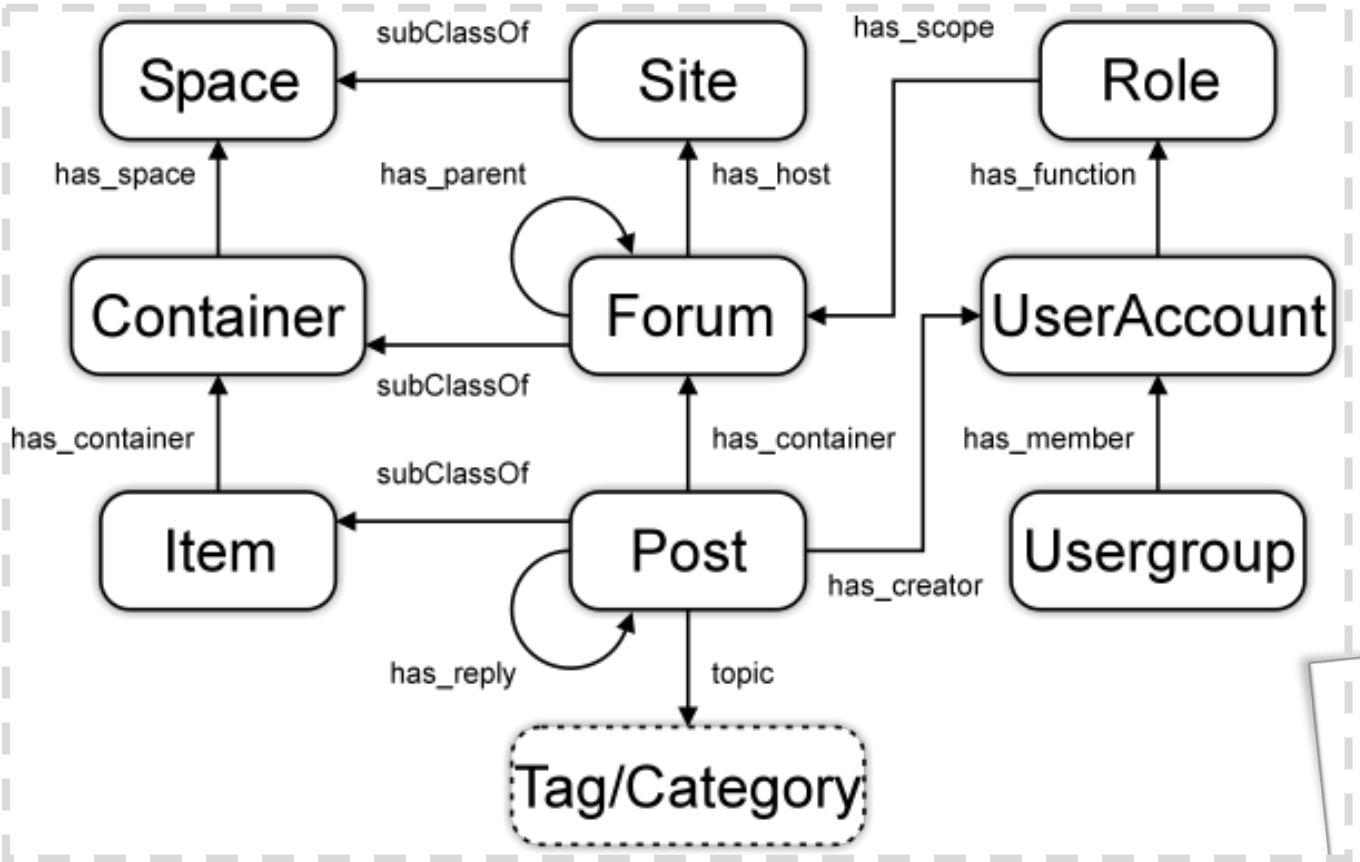
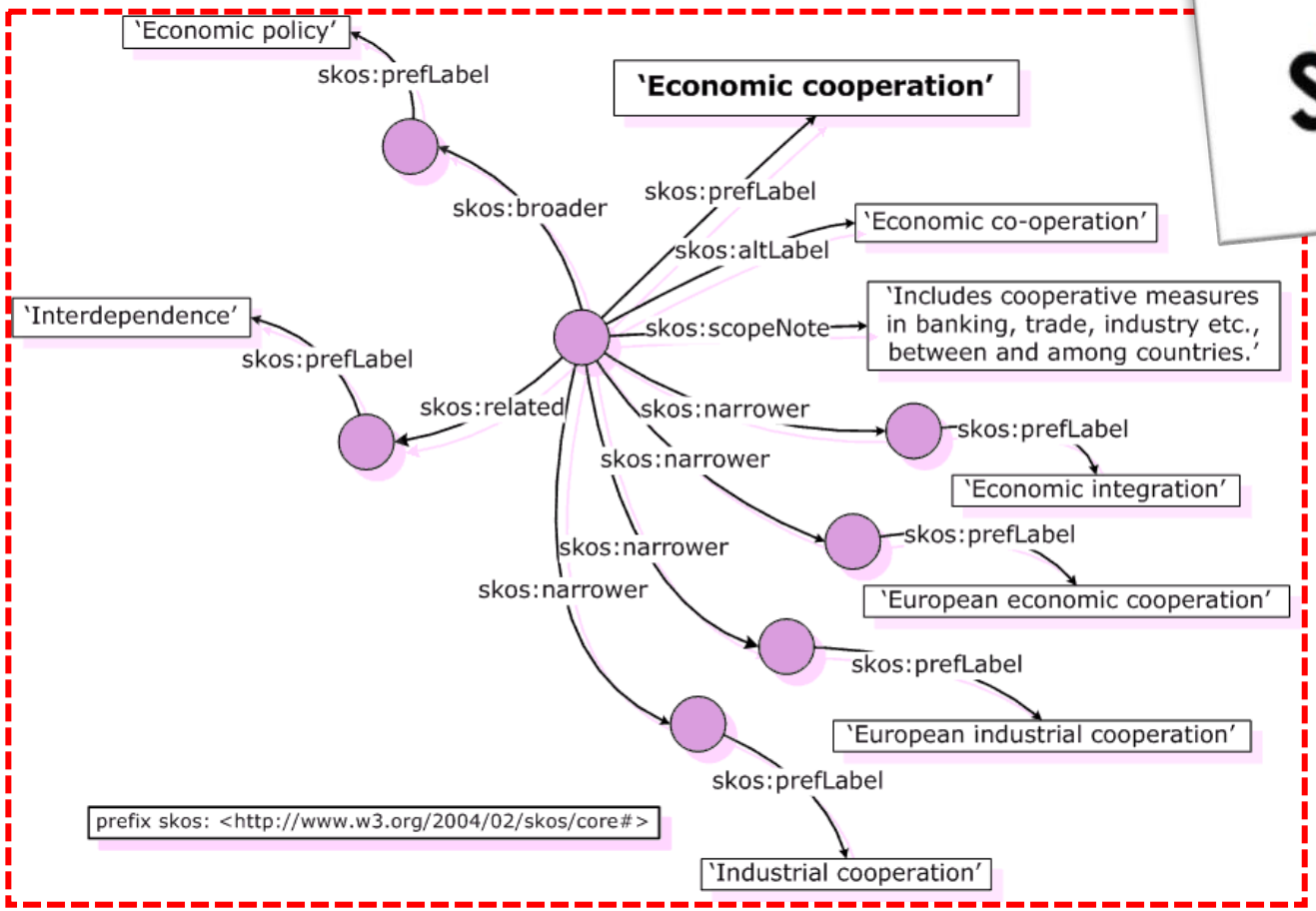


Image from http://rdfs.org/sioc/spec/ : Bojārs, Breslin et al.

Simple Knowledge Organization System



W3C Recommendation

Image from http://www.w3.org/TR/swbp-skos-core-guide. Miles, Brickley

FOAF+SIOC+SKOS

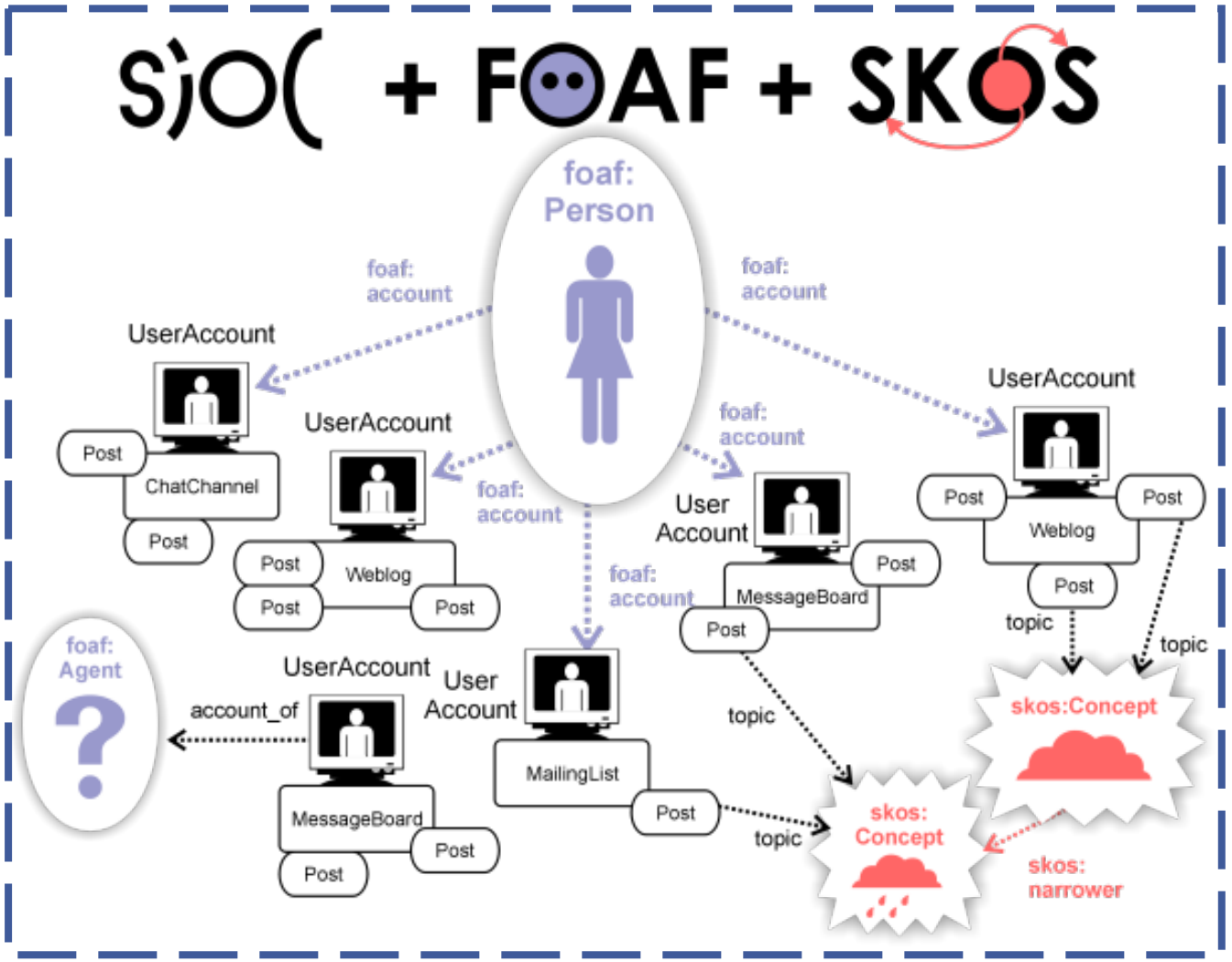


Image from http://sioc-project.org/node/158; **Breslin**

Description Of A Project

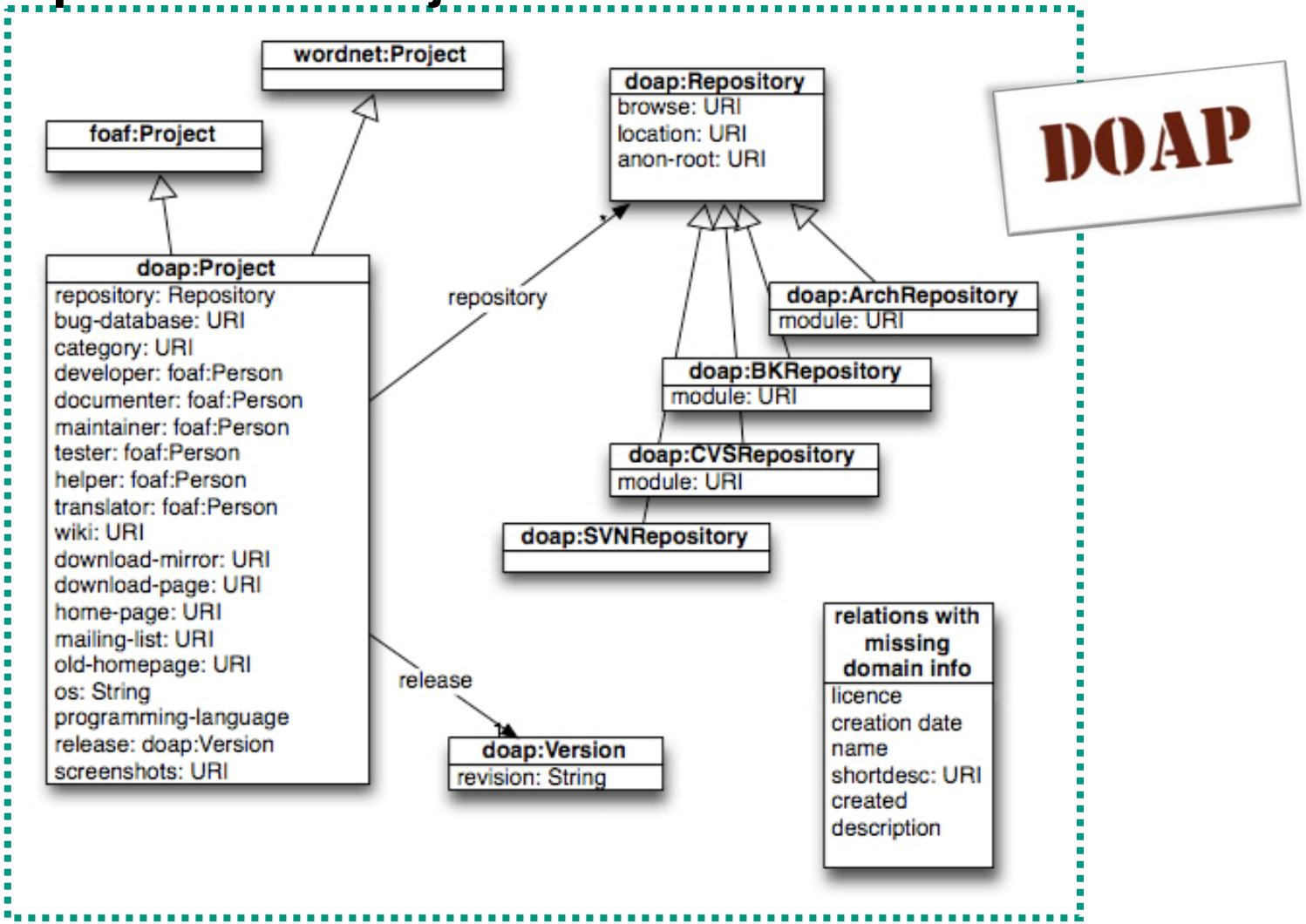


Image from http://code.google.com/p/baetle/wiki/DoapOntology ; **Breslin**

Music Ontology

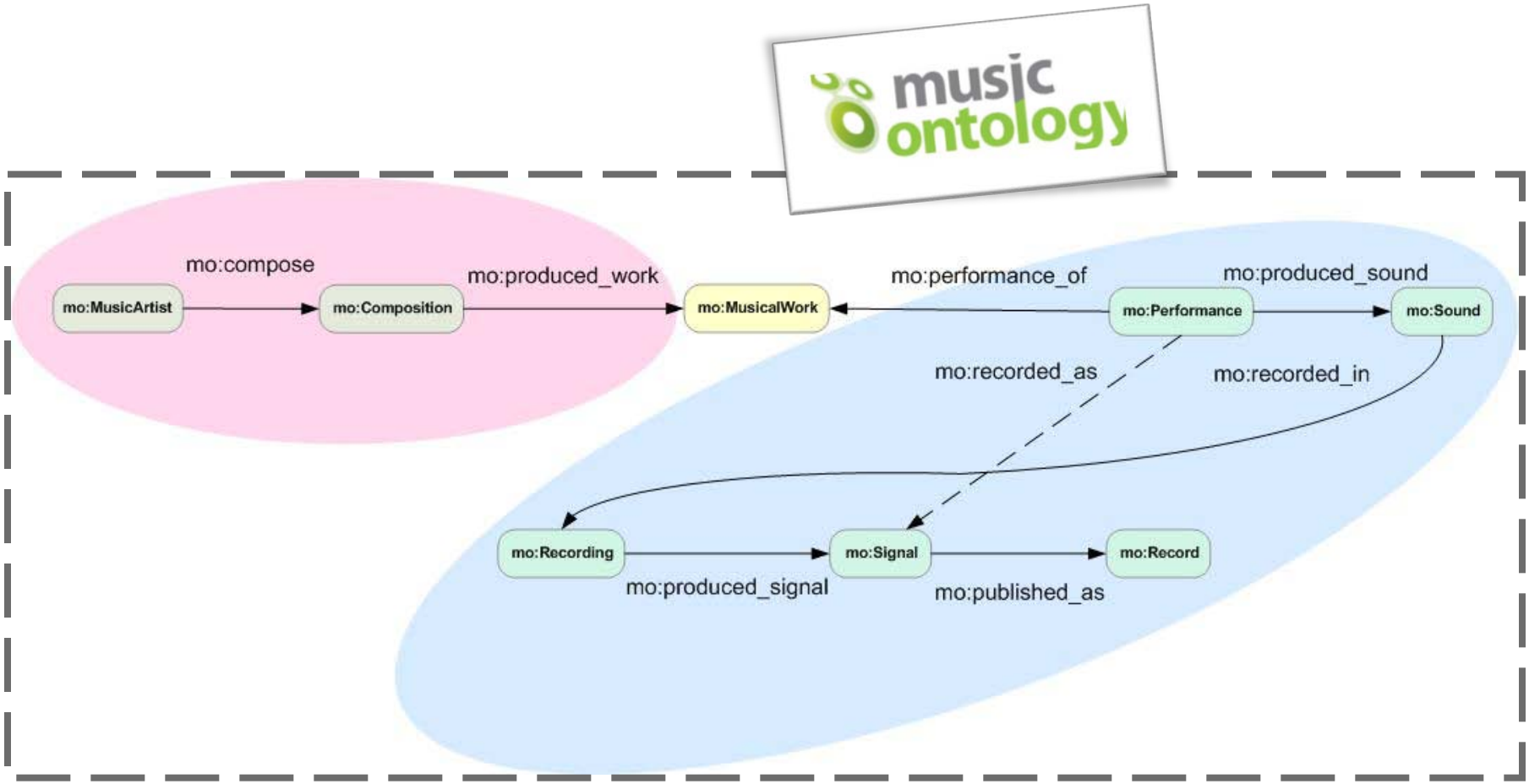


Image from http://musicontology.com/; Raimond, Giasson

GoodRelations

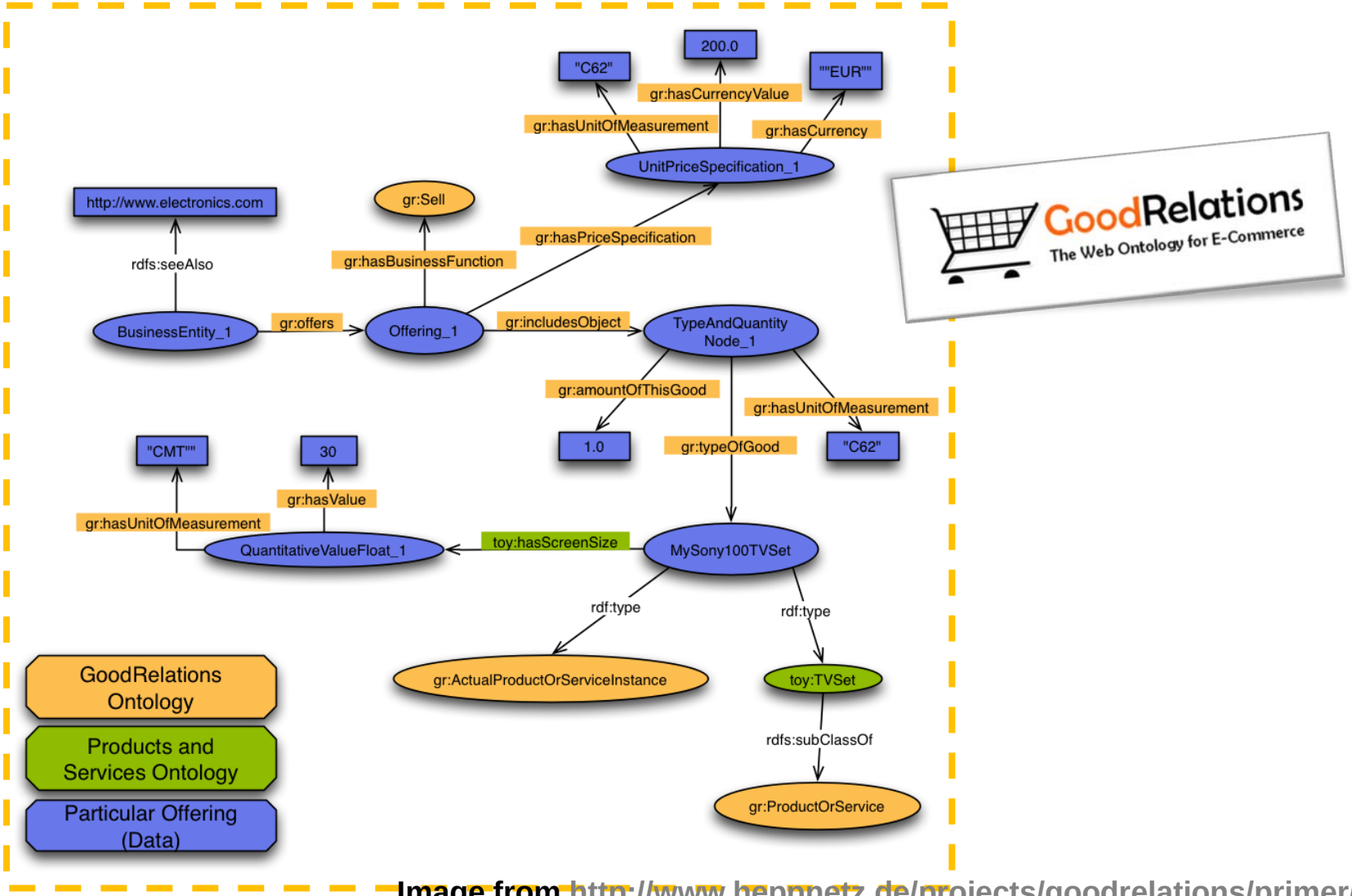
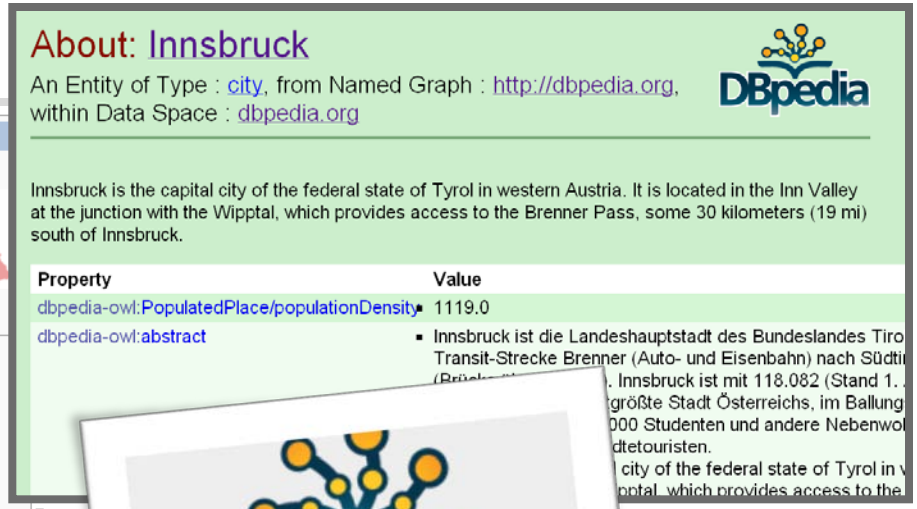


Image from http://www.heppnetz.de/projects/goodrelations/primer/; Hepp

DBpedia

- **Classes and properties for Wikipedia export (infoboxes)**
  - **Cross-domain**
  - 272 classes
  - 1,300 properties

```
{{Infobox Town AT |
  name = Innsbruck |
  image_coa =  InnsbruckWappen.png |
  image_map = Karte-tirol-I.png |
  state = [[Tyrol]] |
  regbzk = [[Statutory city]] |
  population = 117,342 |
  population_as_of = 2006 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 574 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_hem = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_hem = E |
  postal_code = 6010-6080 |
  area_code = 0512 |
  licence = I |
  mayor = Hilde Zach |
  website = [http://innsbruck.at] |
}}
```

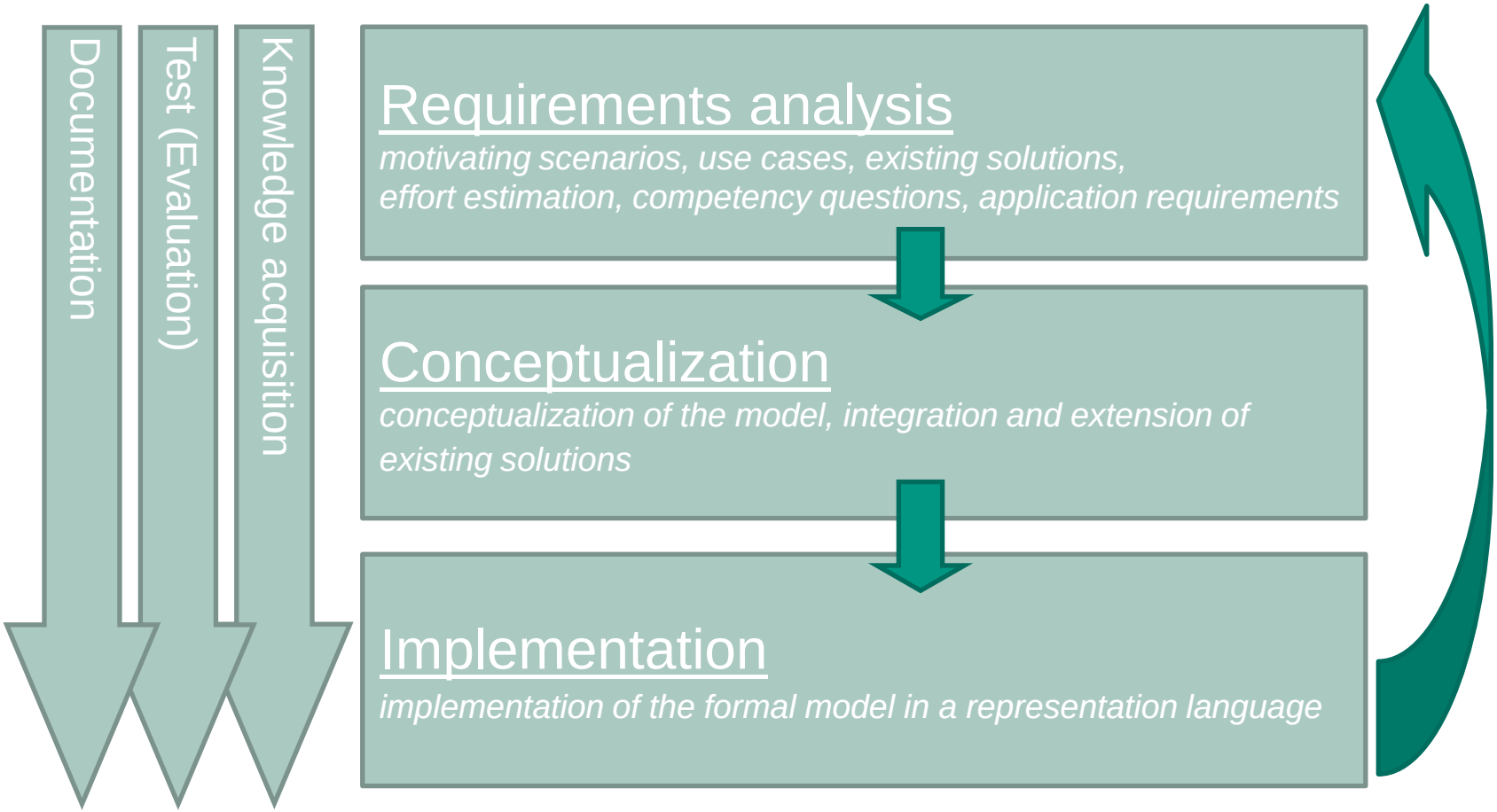


See http://wiki.dbpedia.org/

Natalya F. Noy and Deborah L. McGuinness. ``Ontology Development 101: A Guide to Creating Your First Ontology''. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

HOW TO BUILD AN ONTOLOGY

Process overview



Requirements analysis (1): Domain and scope

- What is the ontology going to be used for?
- Who will use the ontology?
- How it will be maintained and by whom?
- What kind of data will refer to it? And how will these references be created and maintained?
- Are there any information sources available that could be reused?
- What questions should the ontology be able to answer?

- To answer these questions, talk to domain experts, users, and software designers
  - Domain experts don‘t need to be technical, they need to know about the domain, and help you understand its subtleties
  - Users teach you about the terminology that is actually used and the information needs they have
  - Software designers tell you tell you about the type of use cases you need to handle, including the data to be described via the ontology

Requirements analysis (2): Domain vs task-oriented ontologies

- Domain-oriented
  - Ontology models the types of entities in the domain of the application
    - Example: content and features of movies, points of interest in a city, different types of digital camera's…
  - Cover the terminology of the application domain
    - Example: classifications, taxonomies, folksonomies, text corpora
  - Used for annotation and retrieval.
- Task-oriented
  - Ontology serves a purpose in the context of an application
    - Example: finding movies with certain features, recommending sightseeing tours matching my interests, finding and comparing products matching user preferences
  - Define the structure to a knowledge base that can be used to answer competency questions
  - Used for automated reasoning and querying

Requirements analysis (3): Competency questions

- A set of queries which place demands on the underlying ontology
- Ontology must be able to represent the questions using its terminology and the answers based on the axioms
- Ideally, in a staged manner, where consequent questions require the input from the preceeding ones
- A rationale for each competency question should be given

Requirements analysis (4): Finding existing ontologies

- Where to find ontologies
  - Swoogle: over 10 000 documents, across domains
    - http://swoogle.umbc.edu/
  - Protégé Ontologies: several hundreds of ontologies, across domains
    - http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#OWL_ontologies
  - Open Ontology Repository: work in progress, life sciences, but also other domains
    - http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository
  - Tones: 218 ontologies, life sciences and core ontologies.
    - http://owl.cs.manchester.ac.uk/repository/browser
  - Watson: several tens of thousands of documents, across domains
    - http://watson.kmi.open.ac.uk/Overview.html
  - Talis repository
    - http://schemacache.test.talis.com/Schemas/
  - Ontology Yellow Pages: around 100 ontologies, across domains
    - http://wg.sti2.org/semtech-onto/index.php/The_Ontology_Yellow_Pages
  - OBO Foundation Ontologies
    - http://www.obofoundry.org/
  - AIM@SHAPE
    - http://dsw.aimatshape.net/tutorials/ont-intro.jsp
  - VoCamps
    - http://vocamp.org/wiki/Main_Page

Requirements analysis (5): Selecting relevant ontologies

- What will the ontology be used for?
  - Does it need a natural language interface and if yes in which language?
  - Do you have any knowledge representation constraints (language, reasoning)?
  - What level of expressivity is required?
  - What level of granularity is required?
- What will you reuse from it?
  - Vocabulary++.
- How will you reuse it?
  - Imports: transitive dependency between ontologies.
  - Changes in imported ontologies can result in inconsistencies and changes of meanings and interpretations, as well as computational aspects.

Conceptualization (1): Vocabulary

- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?
- Competency questions provide a useful starting point.
- Goint out too far vs. going down too far.
- Investigate homonyms and synonims.

Conceptulization (2): Classes

- Select the terms that describe objects having independent existence rather than terms that describe these objects
  - These terms will be classes in the ontology
- Classes represent concepts in the domain and not the words that denote these concepts
  - Synonyms for the same concept do not represent different classes
- Typically nouns and nominal phrases, but not restricted to them
  - Verbs can be modeled as classes, if the emphasis is on the process as a whole rather than the actual execution

Conceptualization (3): Class hierarchy

- A subclass of a class represents a concept that is a “kind of” the concept that the superclass represents
- It has
  - Additional properties
  - Restrictions different from those of the superclass, or
  - Participates in different relationships than the superclasses
- All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality
- If a class has only one direct subclass there may be a modeling problem or the ontology is not complete
- If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary

- Functional inclusion
  - A chair is-a piece of furniture
  - A hammer is a tool
- State inclusion
  - Polio is a disease
  - Hate is an emotion
- Activity inclusion
  - Tennis is a sport
  - Murder is a crime
- Action inclusion
  - Lecturing is a form of talking
  - Frying is a form of cooking
- Perceptual inclusion
  - A cat is a mammal
  - An apple is a fruit

Conceptualization(4): Properties

- We selected classes from the list of terms in a previous step
  - Most of the remaining terms are likely to be properties of these classes
- For each property in the list, we must determine which class it describes
  - Properties are inherited and should be attached to the most general class in the hierarchy
- Two types of principal characteristics
  - Measurable properties: attributes
  - Inter-class connections: relationships.
    - Use relationships to capture something with an identity
  - Arrest details as attribute of the suspect vs. arrest as an relationship
    - Do we measure degrees of arrestedness or do we want to be able to distinguish between arrests?
  - Color of an image as attribute vs. class
    - A „pointing finger“ rather than a „ruler“ indicates identity

Conceptualization (5): Domain and ranges

- Refine the semantics of the properties
  - Cardinality
  - Domain and range
    - When defining a domain or a range for a slot, find the most general classes or class that can be respectively the domain or the range for the slots
    - Do not define a domain and range that is overly general
    - General patterns for domain and range
      - A class and a superclass – replace with the superclass
      - All subclasses of a class – replace with the superclass
      - Most subclasses of a class – consider replacing with the superclass

Conceptualization (6): Inverse properties

- Modeling with inverse properties is redundant, but
  - Allows acquisition of the information in either direction
  - Enables additional verification
  - Allows presentation of information in both directions

- The actual implementation differs from system to system
  - Are both values stored?
  - When are the inverse values filled in?

Ontology engineering today

- Various domains and application scenarios: life sciences, eCommerce, Linked Open Data
- Engineering by reuse for most domains based on existing data and vocabularies
  - Alignment of data sets
  - Data curation
  - Human-aided computation (e.g., games, crowdsourcing)
- Most of them much simpler and easier to understand than the often cited examples from the 90s
  - However, still difficult to use (e.g., for mark-up)

Open topics

- Meanwhile we have a better understanding of the scenarios which benefit from the usage of semantics and the technologies they typically deploy.
  - Guidelines and how-to‘s
  - Design principles and patterns
  - Schema-level alignment (data-driven)
  - Vocabulary evolution
  - Assessment and evaluation
- Large-scale approaches to knowledge elicitation based on combinations of human and computational intelligence.

Karlsruhe Institute of Technology

Institut AIFB – Angewandte Informatik und Formale Beschreibungsverfahren