NeOn

STLab
The Semantic Technology Lab
ISTC-CNR Rome

Computational Ontologies

Aldo Gangemi
Valentina Presutti
Semantic Technology Lab (ISTC-CNR), Roma
{aldo.gangemi, valentina.presutti}@istc.cnr.it

Credits:
Eva Blomqvist, Sean Bechhofer, Kingsley Idehen, Fabien Gandon, Harry Halpin, Jim Hendler, Tim Berners-Lee

Course Outline

- Day 1: Ontology Languages, Tools, and the Semantic Web
- Day 2: Ontology Design and Ontology Design Patterns
- Day 3: Content Ontology Design Patterns
- Day 4: Design by Re-Engineering

- Lectures, hands-on sessions, discussions, and experiments everyday

- Interaction and fun are key in this course (cf. SSSW)

NeOn experiments

- NeOn is a 14 M€ Integrating Project of EU FP6, 2006-2010
  - www.neon-project.org
  - www.neon-toolkit.org
- We coordinate the work package on Collaborative Ontology Design (C-OD)
  - this PhD Course is funded by NeOn, and some experiments related to best practices in C-OD will be submitted to you
- Started collaborative design repository
  - www.ontologydesignpatterns.org

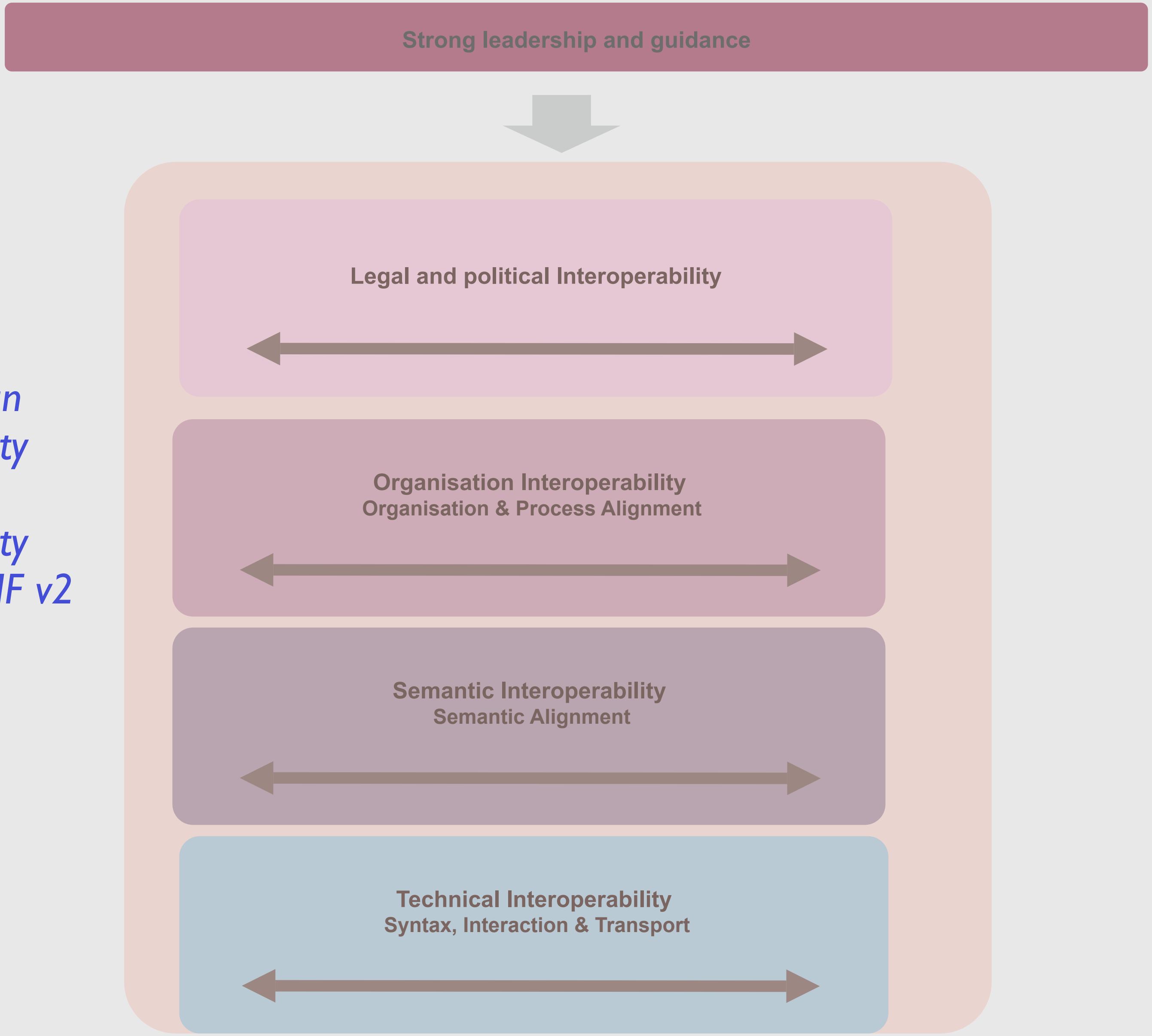
Course Outline

- Ontology Languages, Tools, and the Semantic Web
- Ontology Design and Ontology Design Patterns
- Content Ontology Design Patterns
- Design by Re-Engineering

The motivations for semantic technologies

- Semantic dimension of communities (and of organizations where they operate)
- Heterogeneous contents, distributed and hardly interoperable
- Information extraction, personalized views, and content integration are massive challenges
- Bottlenecks in the interoperability between agents and systems
    - *linguistic* (linguistic structures are for humans, not machines)
    - *contextual* (relational nature of meaning)
    - *social* (information structures not designed for distributed cooperation)

The European Interoperability Framework



Strong leadership and guidance

Legal and political Interoperability

Organisation Interoperability
Organisation & Process Alignment

Semantic Interoperability
Semantic Alignment

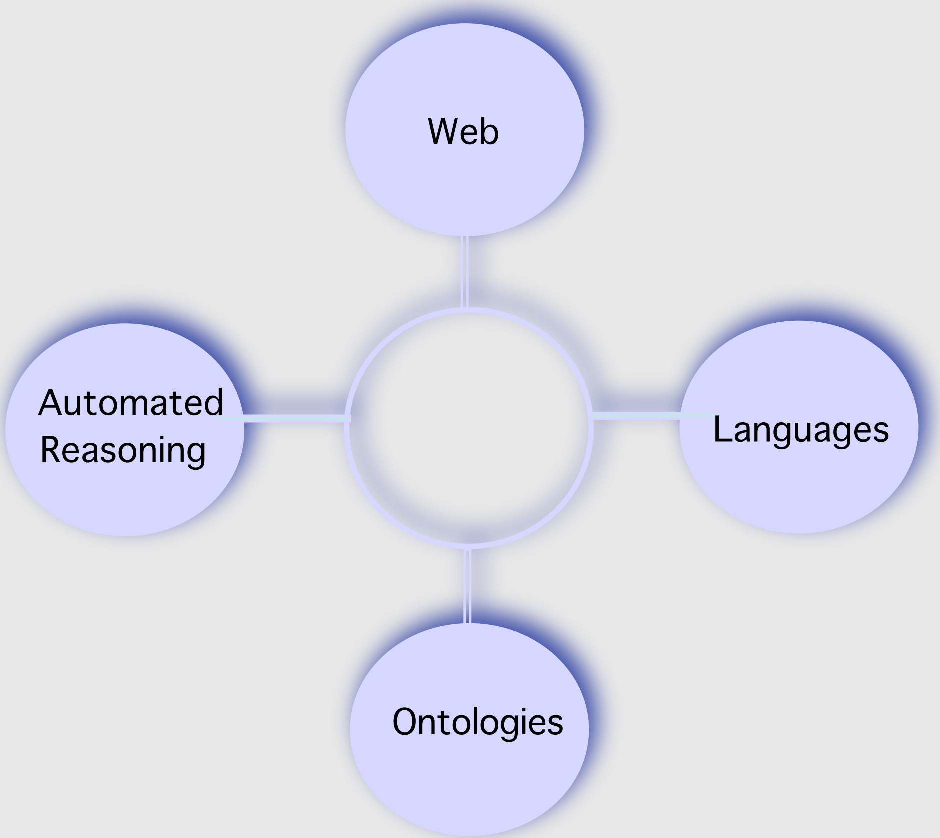
Technical Interoperability
Syntax, Interaction & Transport

The European Interoperability Framework Interoperability dimensions in EIF v2

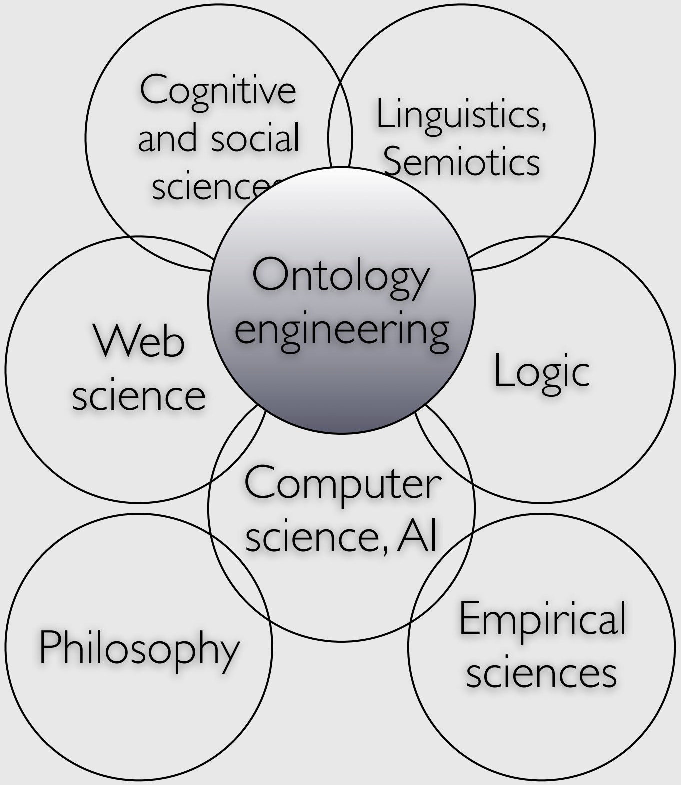
Opportunities

- Semantic web methods and technologies attack the semantic dimension of communities
- NLP methods and technologies tackle the linguistic behavior of communities and organizations
- Conceptual modeling and ontology engineering methods allow the formal description of the structure underlying communities and organizations

A *hub* for web semantics

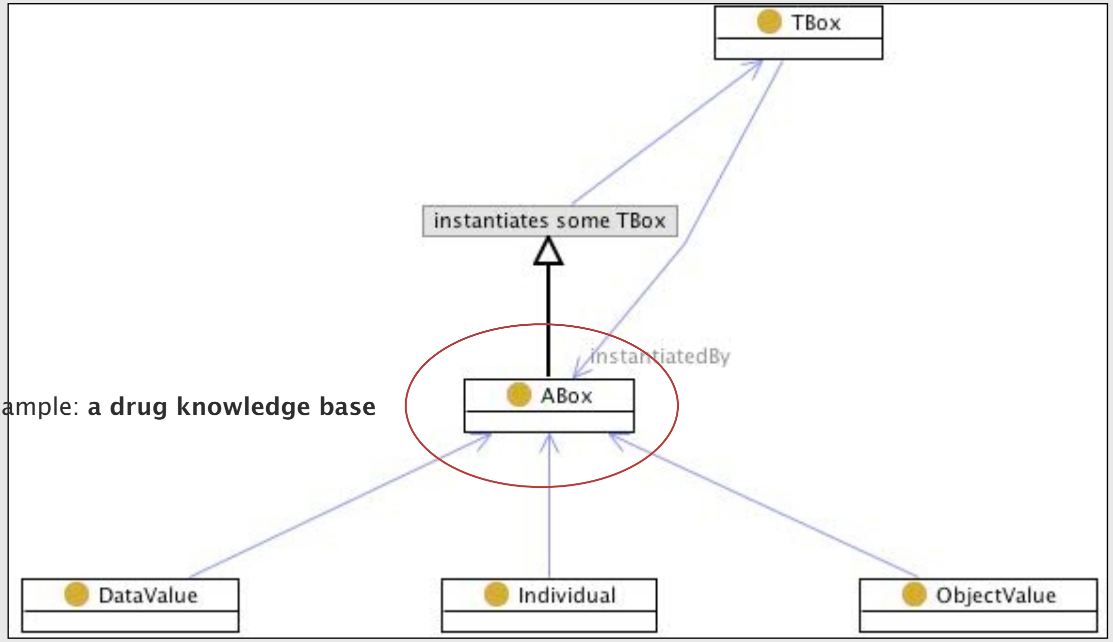


The cultural context of computational ontologies



The elements of an ontology: ABox

Example: **a drug ontology** without instantiations



Example: **a drug knowledge base**

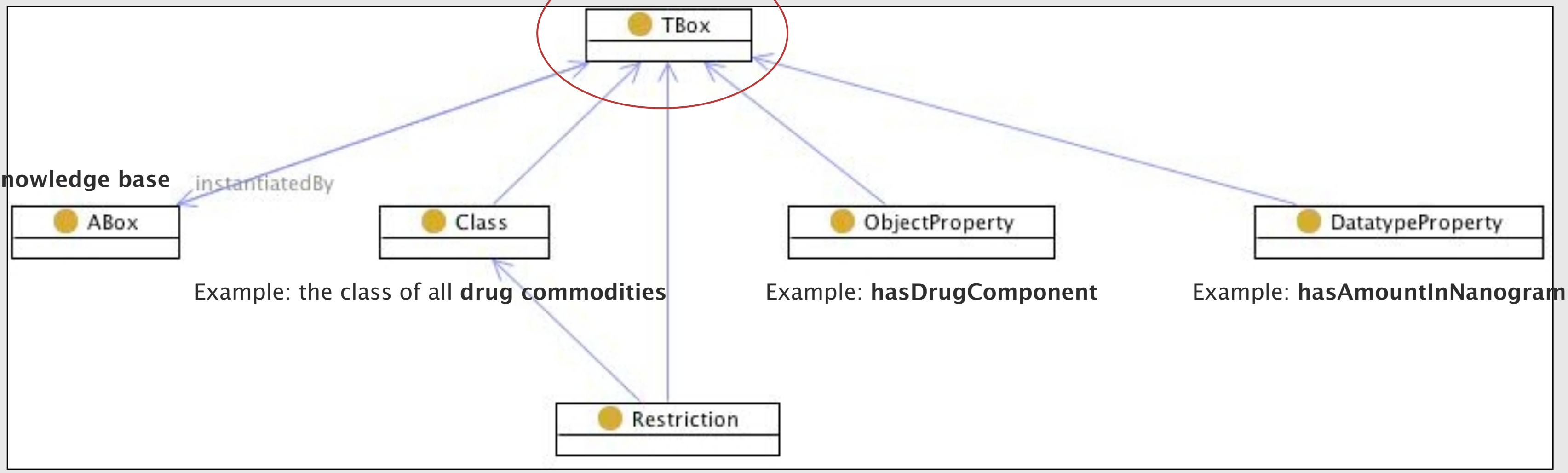
Example:
hasAmountInNanograms : 234

Example: **AcetylsalicylicAcid**,
or **an amount of substance** of that type,
or **a packaged commodity containing it as a component**

Example:
hasDrugComponent : AcetylsalicylicAcid

The elements of an ontology: TBox

Example: a **drug ontology** without instantiations



Example: a **drug knowledge base**

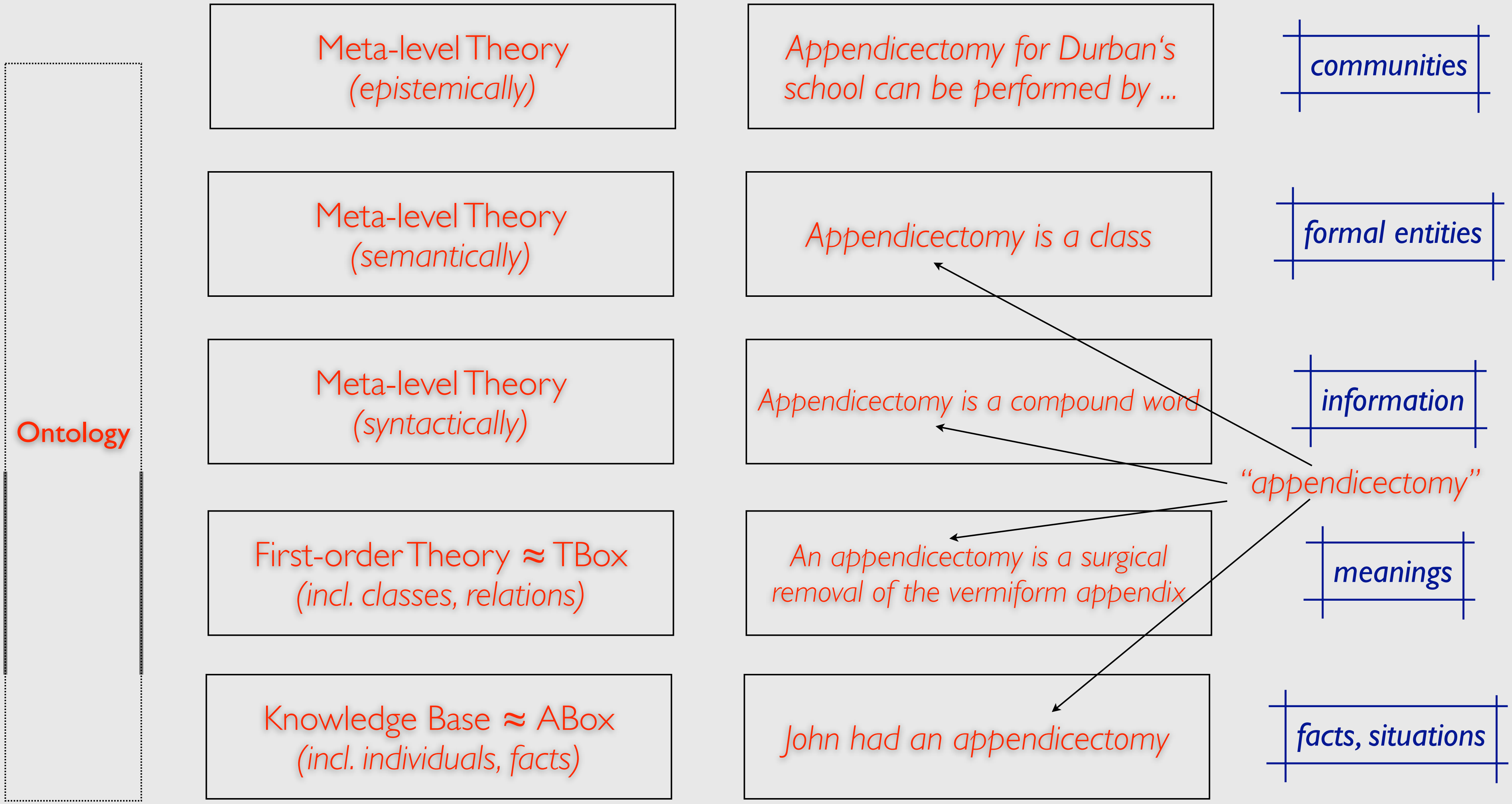
Example: the class of all **drug commodities**

Example: **hasDrugComponent**

Example: **hasAmountInNanograms**

Example: AspirinCommodity **hasDrugComponent 'some' AcetylsalicylicAcid**

Logical layers, types of entities, and contexts



Ontology

Meta-level Theory
(epistemically)

Appendicectomy for Durban's school can be performed by ...

communities

Meta-level Theory
(semantically)

Appendicectomy is a class

formal entities

Meta-level Theory
(syntactically)

Appendicectomy is a compound word

information

"appendicectomy"

First-order Theory ≈ TBox
(incl. classes, relations)

An appendicectomy is a surgical removal of the vermiform appendix

meanings

Knowledge Base ≈ ABox
(incl. individuals, facts)

John had an appendicectomy

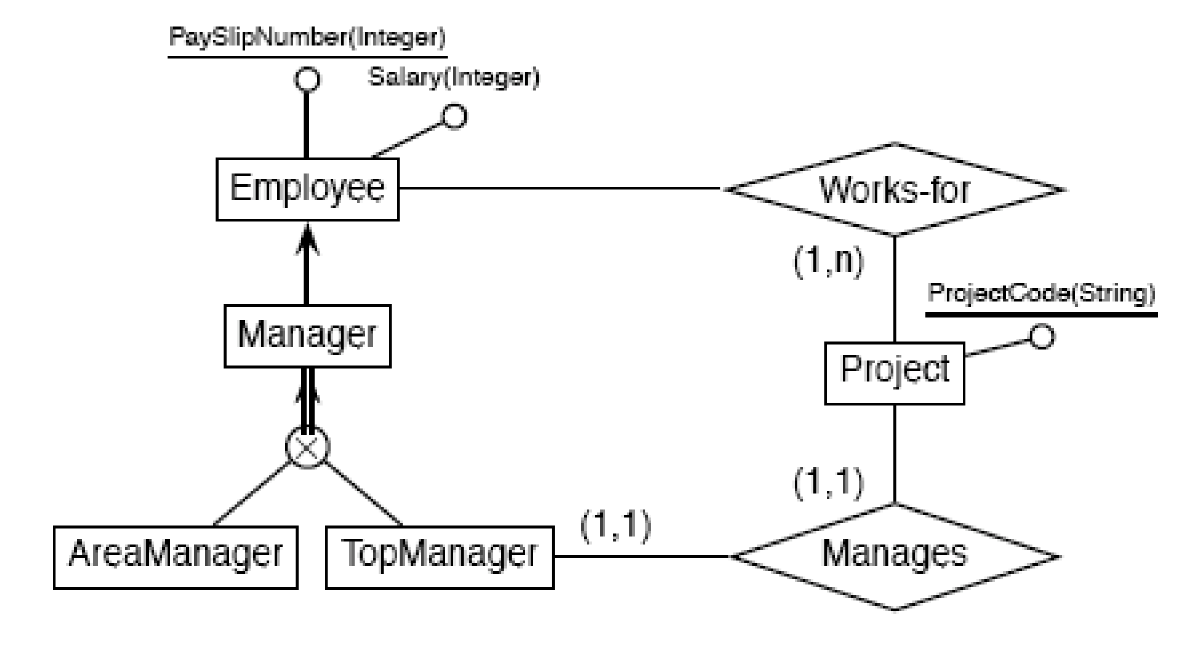
facts, situations

What is ontology design?

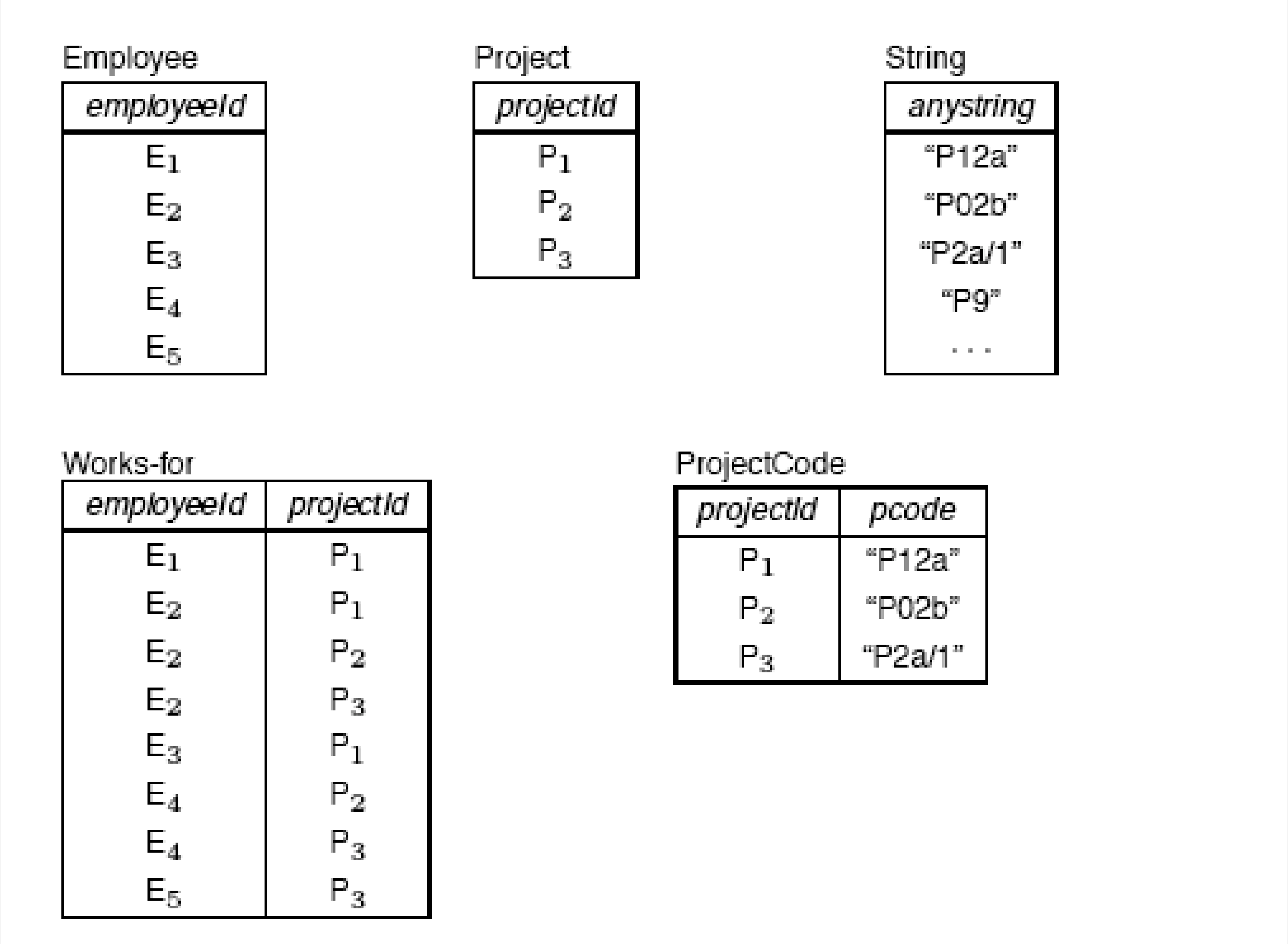
- Ontologies are artifacts
  - Have a structure (linguistic, “taxonomical”, logical)
  - Their function is to “encode” a description of the world (actual, possible, counterfactual, impossible, desired, etc.) for some purpose
- Ontologies must match both domain and task
  - Allow the description of the entities (“domain”) whose attributes and relations are concerned by some purpose, e.g. *drugs as commodities that contain preparations of selected compounds having an expected application within medical treatments*
  - Serve a purpose (“task”), e.g. *finding piperocaine-based anesthetic drugs*, *integrating a drug database with a compound database*, *matching available resources to devised drug production plans*, etc.
- Ontologies have a lifecycle
  - Are created, evaluated, fixed, and exploited just like any artifact
  - Their lifecycle has some original characteristics regarding:
    - *Data*
    - *Project* and *workflow types*
    - *Argumentation structures*
    - *Design patterns*

Data modeling: Entity/Relationship

- Used for modeling the domain
- Typically mapped to a relational representation
- Similar to an ontology



Relational Representation



Employee

employeeId
E_1
E_2
E_3
E_4
E_5

Project

projectId
P_1
P_2
P_3

String

anystring
“P12a”
“P02b”
“P2a/1”
“P9”
. . .

Works-for

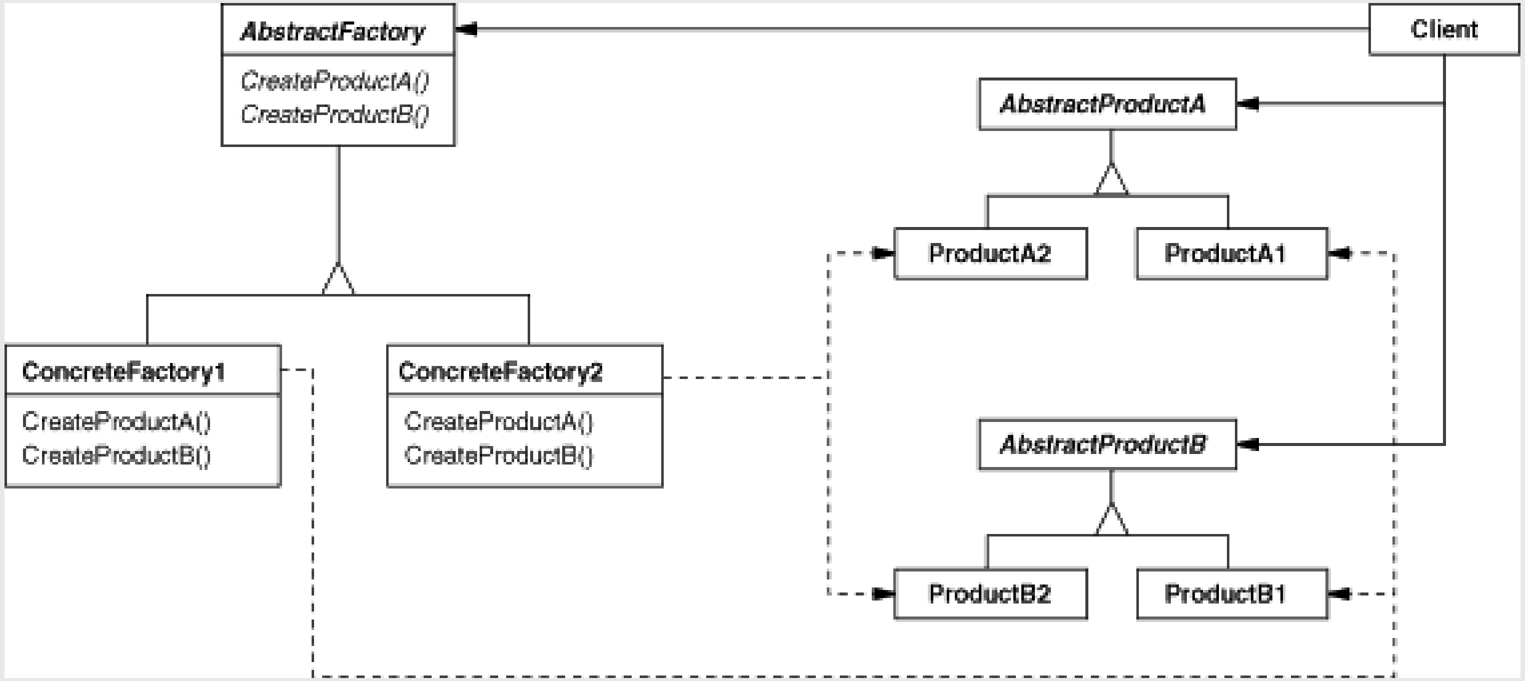
employeeId	*projectId*
E_1	P_1
E_2	P_1
E_2	P_2
E_2	P_3
E_3	P_1
E_4	P_2
E_4	P_3
E_5	P_3

ProjectCode

projectId	*pcode*
P_1	“P12a”
P_2	“P02b”
P_3	“P2a/1”

OO modeling

- Typically it is used to model software solutions
- Classes have a behavior (procedural aspects)



What is needed for designing ontologies

- Resources ("raw" material)
  - *Reengineering* is key
  - Thesauri2ABox, Lexicon2TBox, Tags2ABox, etc.

cf. Course 4

- Formal languages

cf. Course 1

- Solutions (target configurations for the raw material)
  - *Design patterns* are key
  - Logical, architectural, reasoning, and content design

cf. Course 2 and 3

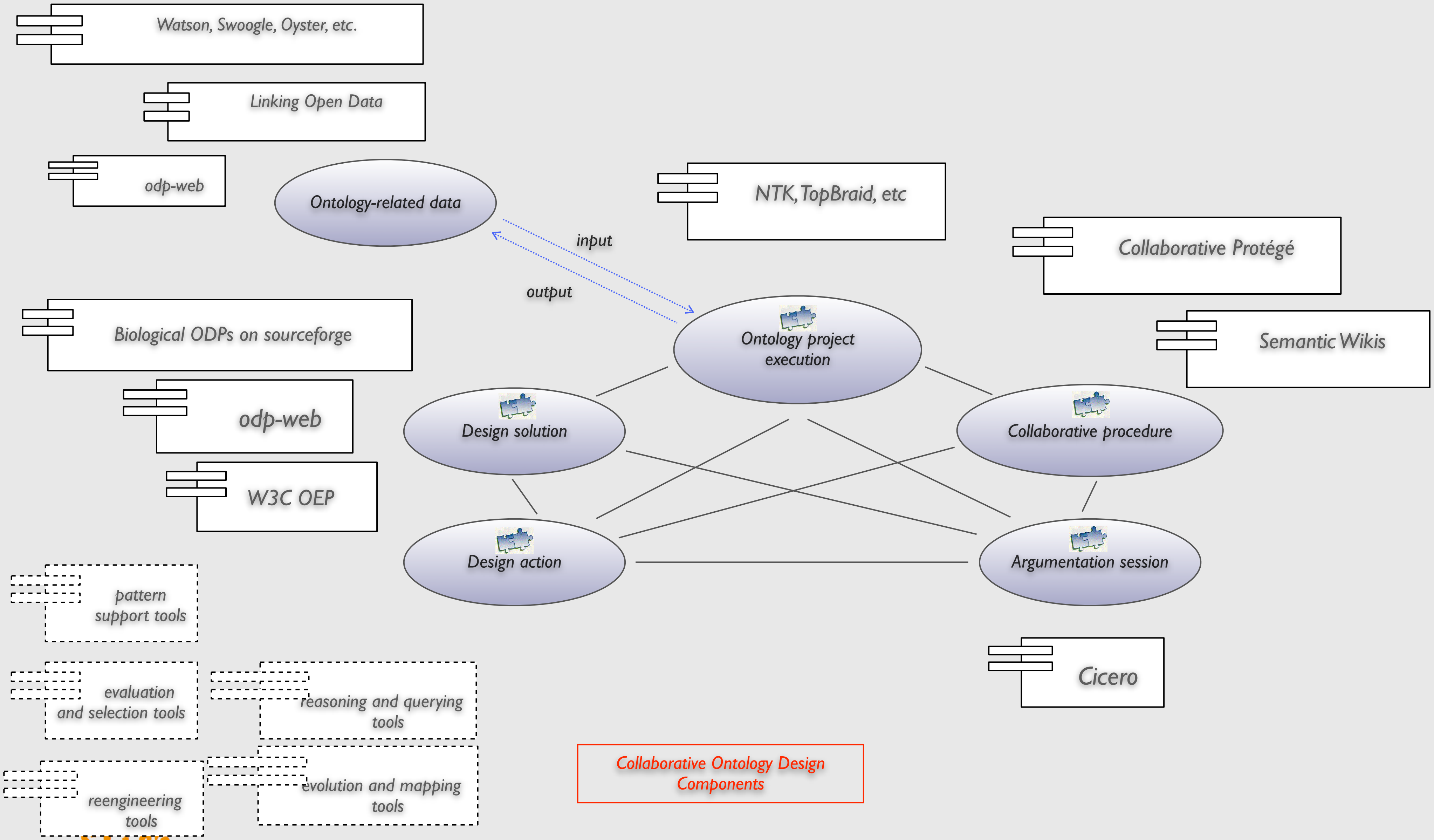
- Methods (production from raw material)
  - *Collaboration workflows* are key
    - Lone wolf, small research team, hierarchical commercial team, dictator
  - Argumentation, search, evaluation, selection, reengineering procedures, pattern matching and composition

cf. Course 2 and 4 (partly)

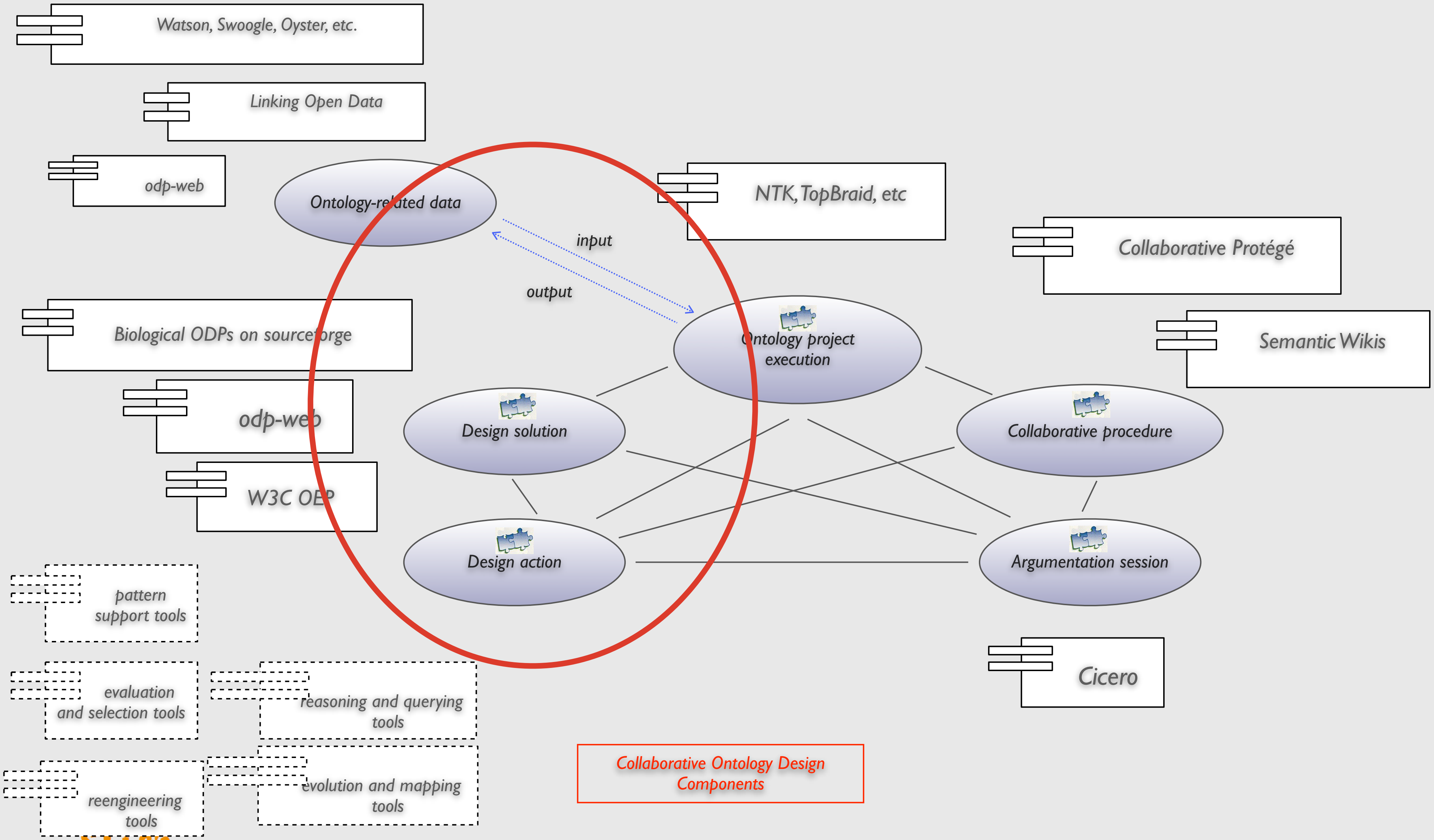
- Tools that implement methods
  - Current state not yet satisfactory, but improving (TopBraid, Protégé4, NeOn Toolkit, ...)

cf. Hands-on sessions

Design in C-ODO



Design in C-ODO



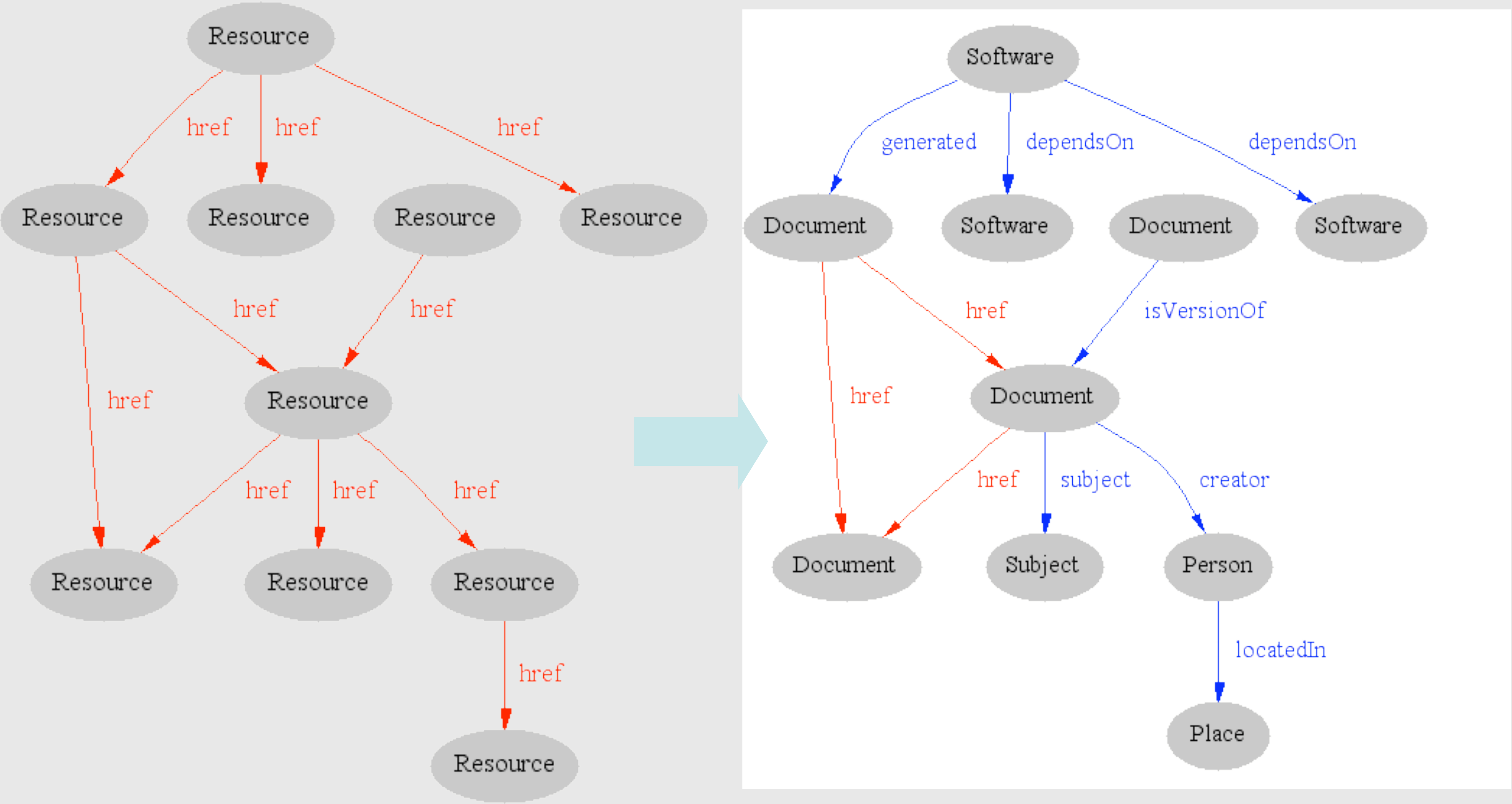
Ontology-related data

- “Knowledge resource” is a role for several kinds of information objects
- Informal vs. formal
  - Text corpora
  - Folksonomies (tag sets, directories, topic trees, subject indexes, infoboxes)
  - Lexica (dictionaries, wordnets, terminologies, nomenclatures)
  - Knowledge organization systems (thesauri, classification schemes)
  - Frames, semantic networks
  - DB schemas
  - Linked Open Data datasets
  - (Computational) ontologies
- Suppose we need to design an ontology of *desire* ... where to start from?

Semantic Web and Web ontologies

- Make web resources more accessible to automated processes
- Extend existing rendering markup with semantic markup
  - Metadata annotations that describe content/function of web accessible resources
- Use ontologies to provide vocabulary for annotations
  - New terms can be formed by combining existing ones
  - “Formal specification” is accessible to machines
- A prerequisite is a standard web ontology language
  - Need to agree on a common syntax before we can share semantics
  - Syntactic web based on standards such as HTTP and HTML

Web vs. Semantic Web



Resources for the Semantic Web

- Metadata
  - Resources are marked-up with descriptions of their content. No good unless everyone speaks the same language
- Terminologies
  - provide shared and common vocabularies of a domain, so search engines, agents, authors and users can communicate. No good unless everyone means the same thing
- Ontologies
  - provide a shared and common understanding of a domain that can be communicated across people and applications, and will play a major role in supporting information exchange and discovery

Web Languages

- Web languages already extended to facilitate content description
  - XML Schema (XMLS)
  - RDF and RDF Schema (RDFS)

- RDFS recognizable as an ontology language
  - Classes and properties
  - Range and domain
  - Sub/super-classes (and properties)

RDF

- RDF stands for Resource Description Framework

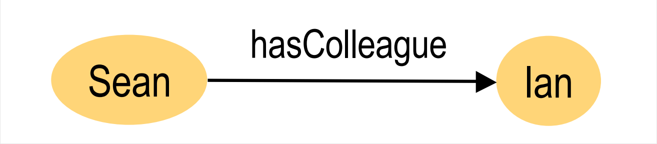
- It is a W3C Recommendation
  - http://www.w3.org/RDF

- RDF is a graphical formalism (+ XML syntax + semantics)
  - for representing metadata
  - for describing the semantics of information in a machine-accessible way

- Provides a simple data model based on triples.

RDF Data Model

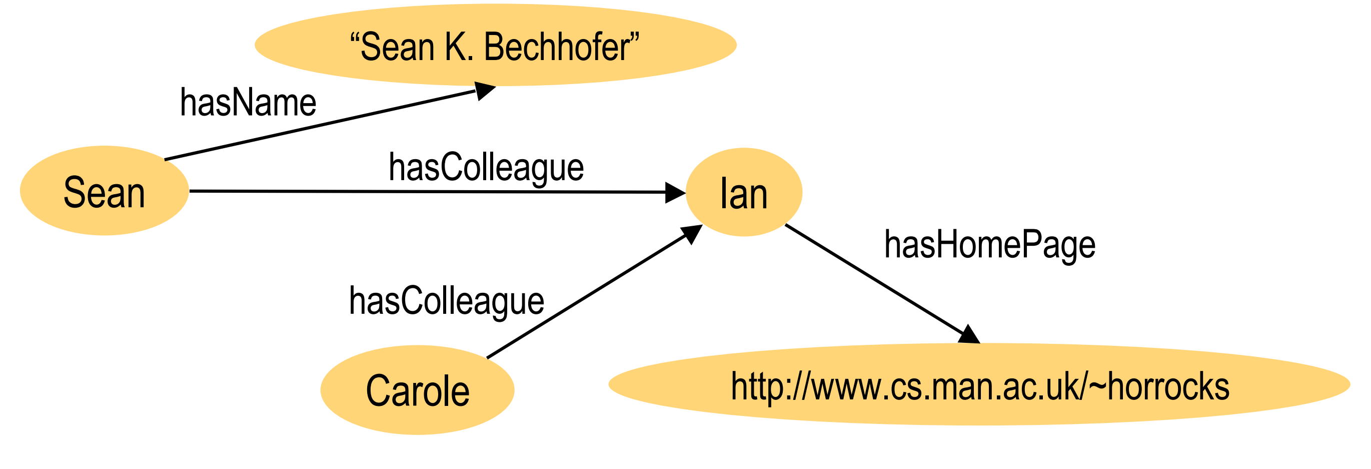
- Statements are <subject, predicate, object> triples:
  - <Sean,hasColleague,Ian>
- Can be represented as a graph:



- Statements describe properties of resources
- A resource is any object that can be pointed to by a URI:
  - The generic set of all names/addresses that are short strings that refer to resources
  - a document, a picture, a paragraph on the Web, http://www.cs.man.ac.uk/index.html, a book in the library, a real person (?), isbn://0141184280
- Properties themselves are also resources (URIs)

Linking Statements

- The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph

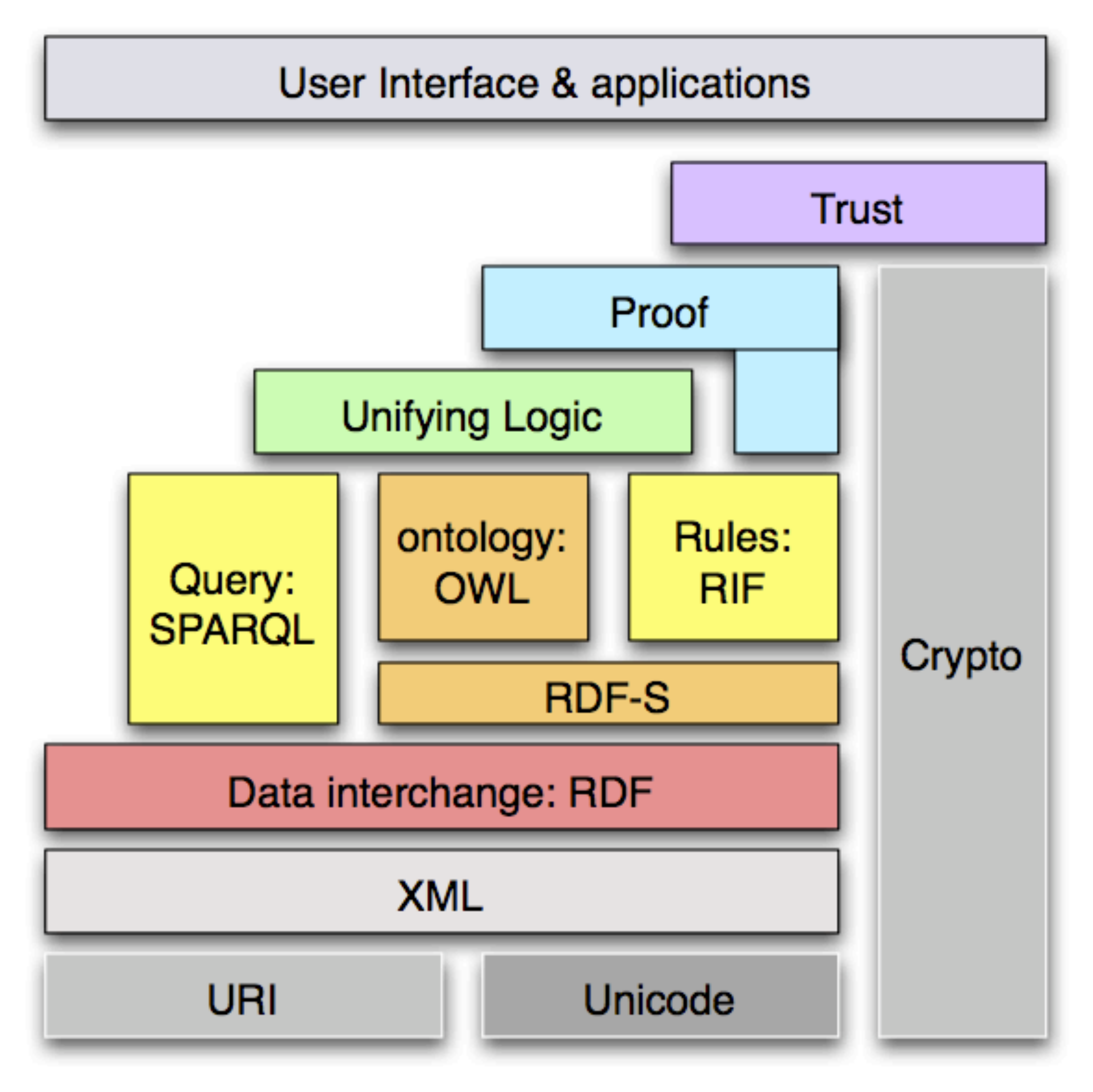


- Note that the object of a triple can also be a "literal" (a string)

What does RDF give us?

- A mechanism for annotating data and resources.
- Single (simple) data model.
- Syntactic consistency between names (URIs).
- Low level integration of data.

The Semantic Web Layers



Ontology supporting tools

- Editors
  - NeOn Toolkit, Protégé4, TopBraid Composer, Swoop, Protégé 3.4, and Collaborative Protégé; they usually aggregate:
    - Ontology Project manager
    - Workspace manager
    - Ontology Browser
    - Ontology Editor
    - Grapher
    - Importer
    - Reasoner engine
    - Query engine
- Reasoners
  - FaCT++, Pellet, Racer, ...
  - F-OWL, E-Wallet, ...
- APIs and Frameworks
  - Jena, WonderWeb OWL API, Protégé OWL API, OWLIM