Semantic Web Technologies II

SS 2008

16.04.2008

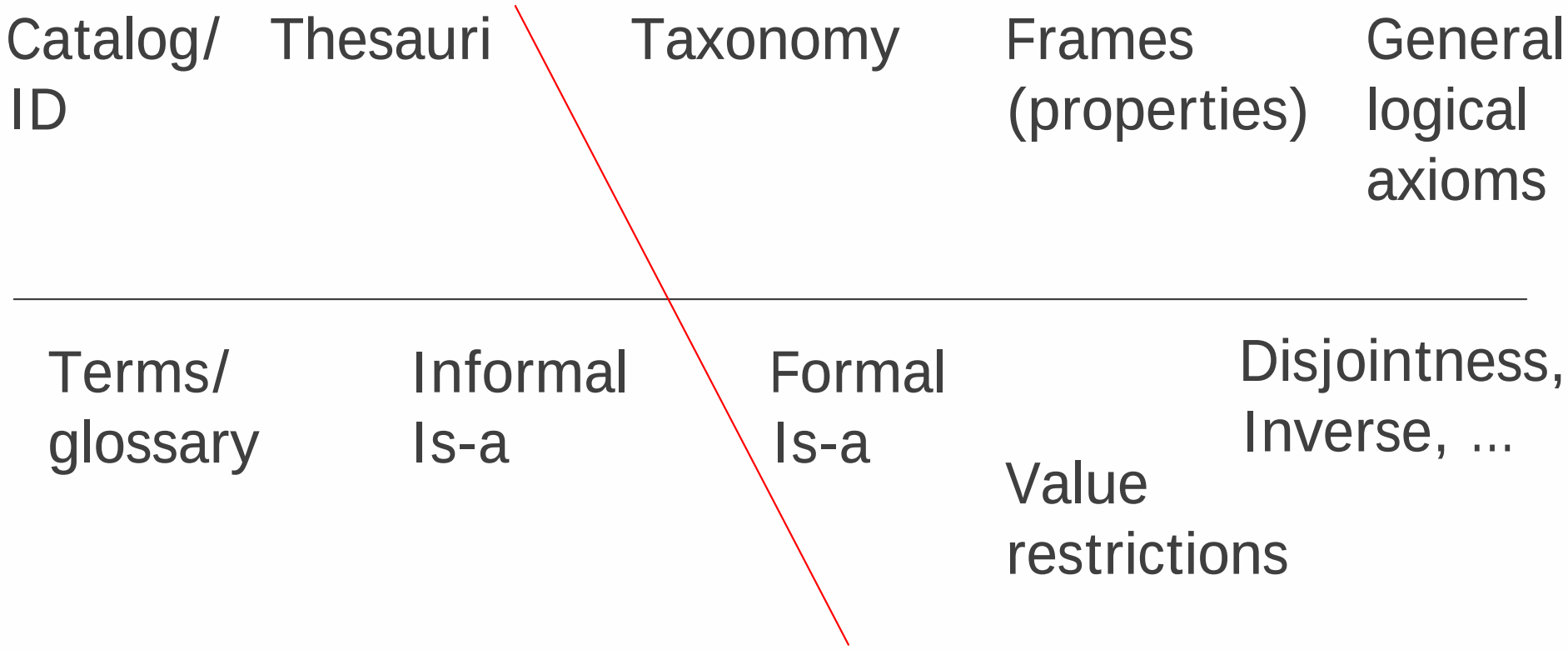
Engineering Ontologies and Semantic Applications

Dr. Peter Haase
PD Dr. Pascal Hitzler
Dr. Steffen Lamparter
Denny Vrandecic

Topics

- What are use cases for ontologies in semantic applications?
- What are development processes and lifecycle activities for engineering
  1) ontologies
  2) semantic applications?
- What are supporting infrastructures/architectures for engineering ontologies and semantic applications?
- What are specific technologies to build semantic applications?

Spectrum of Ontologies



[Deborah McGuinness, Stanford]

Many Ways to Use Ontologies –
A Semantic Continuum

„An ontology is a specification of a conceptualization.“ (Gruber)

Pump: “a device for moving a gas or liquid from one place or container to another”

(pump subclassof device)

Informal **Formal**

(Implicit) (explicit)

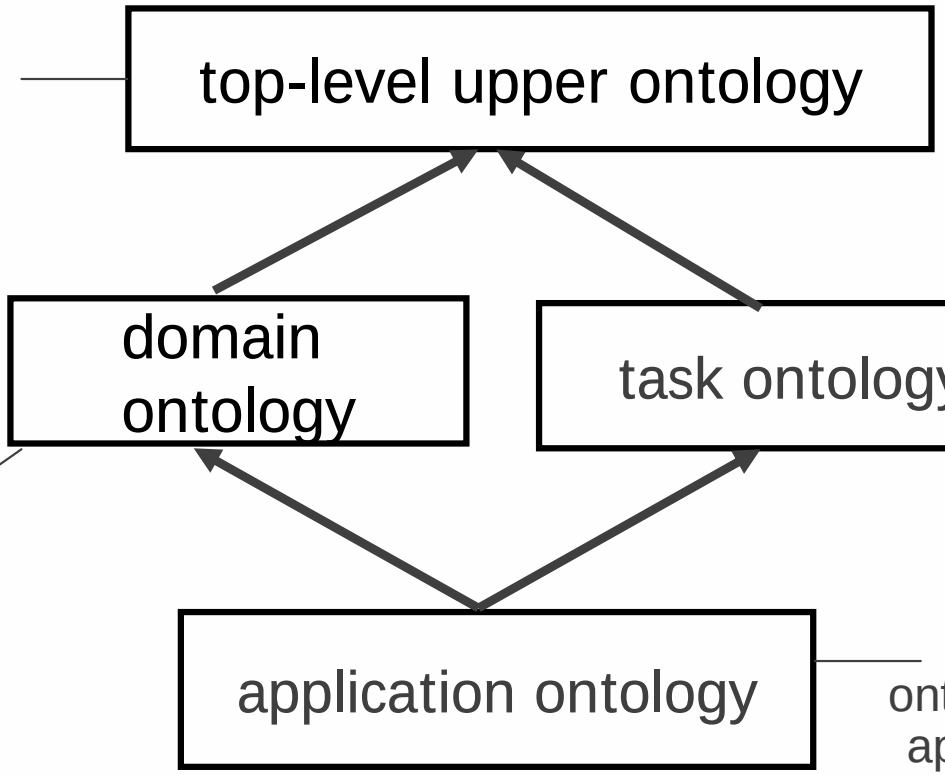
Shared human consensus	Text descriptions	Semantics “hardwired” at runtime	Semantics processed and used at runtime
	e.g. comments to database schema	consistency checking at design time	use of all the axioms at runtime, e.g. for question answering

Further to the right means:

- Less ambiguity
- Better inter-operation
- Less hardwiring
- More robust to change
- More difficult

[Mike Uschold, Boeing Corp]

Coverage/Specify of Ontologies



describe very general concepts like space, time, event, which are independent of a particular problem or domain. It seems reasonable to have unified top-level ontologies for large communities of users.

describe the vocabulary related to a generic task or activity by specializing the top-level ontologies.

describe the vocabulary related to a generic domain by specializing the concepts introduced in the top-level ontology.

the most specific ontologies. Concepts in application ontologies often correspond to roles played by domain entities while performing a certain activity.

[Guarino, 98]

Specific ontologies

- Domain-oriented
  - Domain-specific
    - Medicine => cardiology => rhythm disorders
  - Domain generalizations
    - components, organs, documents, gene function
- Task-oriented
  - task specific
    - configuration design, instruction, planning, annotation analysis
  - task generalizations
    - problem solving methods

Upper Ontologies

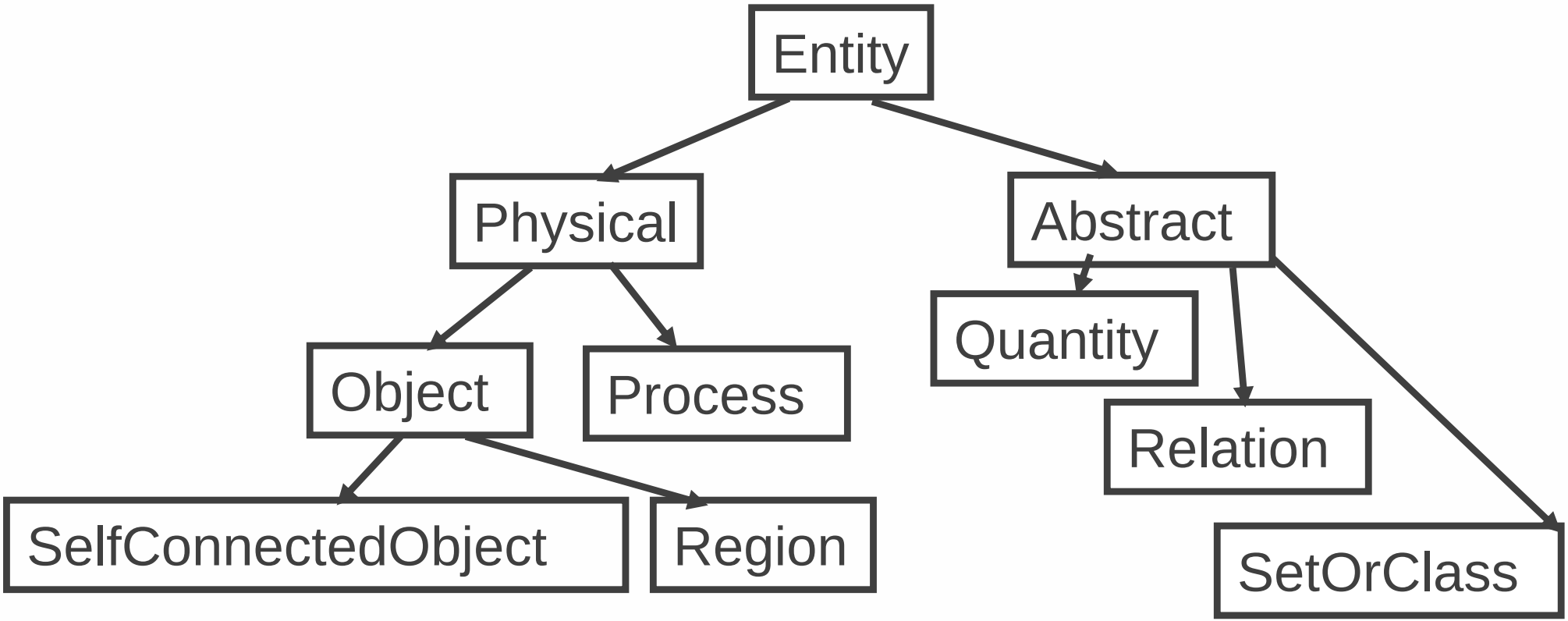
- A.k.a. top level, core, generic or reference
- An attempt to capture the most general and reusable terms and definitions
- “Physical”, “Abstract”, “Structure”, “Substance”
- Useful for ontology re-use
- Important when generating or analysing natural language expressions

- **Examples of Top Level ontologies**
- SUMO
- DOLCE
- CyC
- WordNet
- EuroWordNet

Suggested Upper Merged Ontology (SUMO)

- 1000 terms, 4000 axioms, 750 rules
- Written in FOL
- Development since 2000
- Associated domain ontologies totaling 20,000 terms and 70,000 axioms
- Free
  - SUMO is owned by IEEE but basically public domain
  - Domain ontologies are released under GNU
  - www.ontologyportal.org

SUMO – Top Level Concepts



Cyc

- enCYClopedia
- Douglas Lenat at Cycorp
- Development since 1984
- general knowledge and common-sense reasoning
- Ontology – 100,000's of terms
- Millions of assertions
  - "Water is wet"
  - "Everyone has a mother"
  - "When you let go of things they usually fall."
- Open version available – opencyc.com

WordNet

- "Lexical ontology"
- 100,000 word senses – synsets
- Created by George Miller's group at Princeton
- Free
- De facto standard in the linguistics world

news item IS A KIND OF ...

1 sense of news item

Sense 1
news item -- (an item in a newspaper)
=> **item, point** -- (a distinct part that can be specified separately in a group of things that could be enumerated on a list; "he noticed an item in the New York Times"; "she had several items on her shopping list"; "the main point on the agenda was taken up first")
=> **part, portion, component part, component** -- (something determined in relation to something that includes it; "he wanted to feel a part of something bigger than himself"; "I read a portion of the manuscript"; "the smaller component is hard to reach")
=> **relation** -- (an abstraction belonging to or characteristic of two entities or parts together)
=> **abstraction** -- (a general concept formed by extracting common features from specific examples)

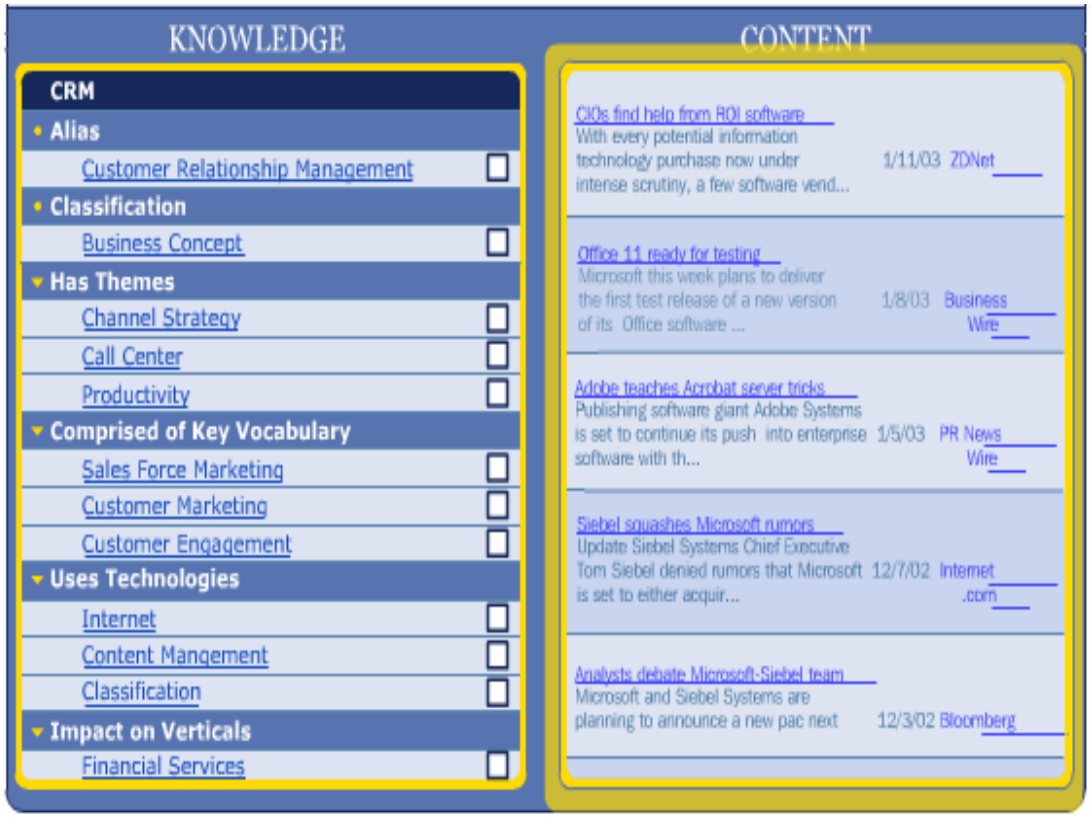
Applications of Ontologies and Semantic Technologies

- Semantic Search
- Information Integration
- Knowledge Management
- Personal Information Management
- Decision Support and Expert Systems
- Semantic Web Services

Example: Semantic Search

1. **Search Term: CRM**

2. **Disambiguation and Navigation of Search Results using Ontologies**



We can use ontologies to enhance search by:

- Sense disambiguation
- Query expansion
- Type with restrictions
- ….

Example: Semantic Search with ORAKEL

"Which journal articles were written by 'Tim Berners-Lee' for which journal?"

```
PREFIX protonu:<http://proton.semanticweb.org/2005/04/protonu#>
PREFIX protont:<http://proton.semanticweb.org/2005/04/protont#>

SELECT ?x ?z WHERE {
   ?x rdf:type protonu:Article .
   ?x protont:documentAuthor ?y .
   ?y rdfs:label ?ys .
   match(?ys, "Tim Berners Lee") .
   ?z rdf:type protonu:Journal .
   ?x protonu:publishedWithin ?z
}
```

"The Semantic Web"	"The Scientific American"
"WWW: Past, Present, and Future"	"IEEE Computer"
[...]	[...]

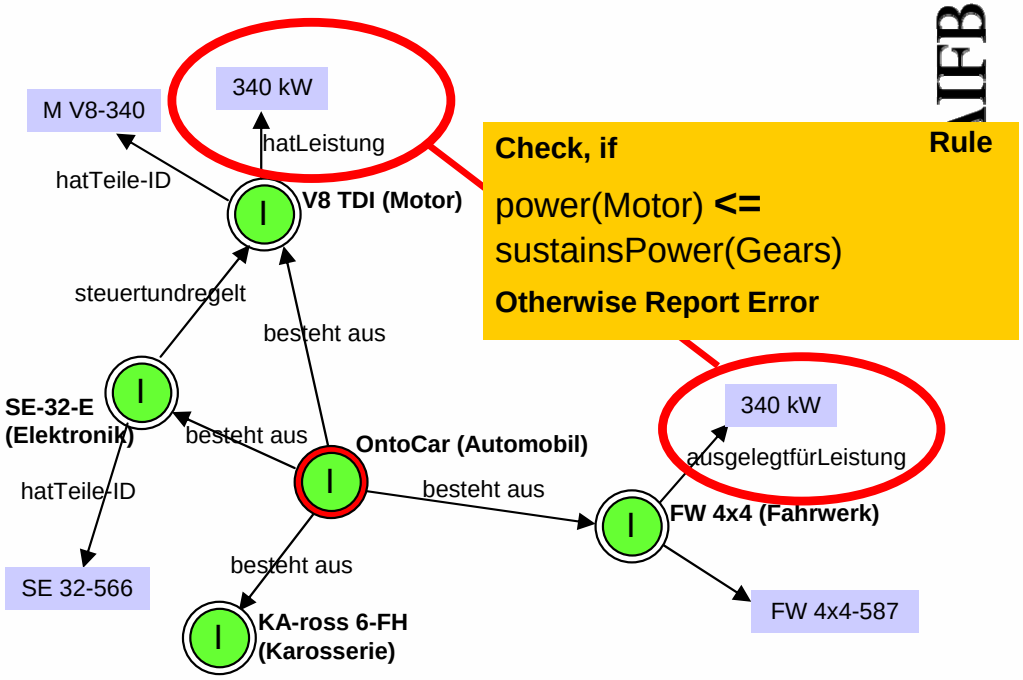
Example: Flexible and Extensible Decision Support Systems

Decision Support Systems (DSS)

- Used to prove correctness of **complex engineering** tasks

Use Case at Audi

- Support of internal preparation of changes to test vehicles (AVx)
- Ontology integrates knowledge of **all** parties involved (several engineering departments)
- Knowledge about **functional, geometric and temporal effects** can be captured in a modular and reusable manner through ontologies and associated rule bases

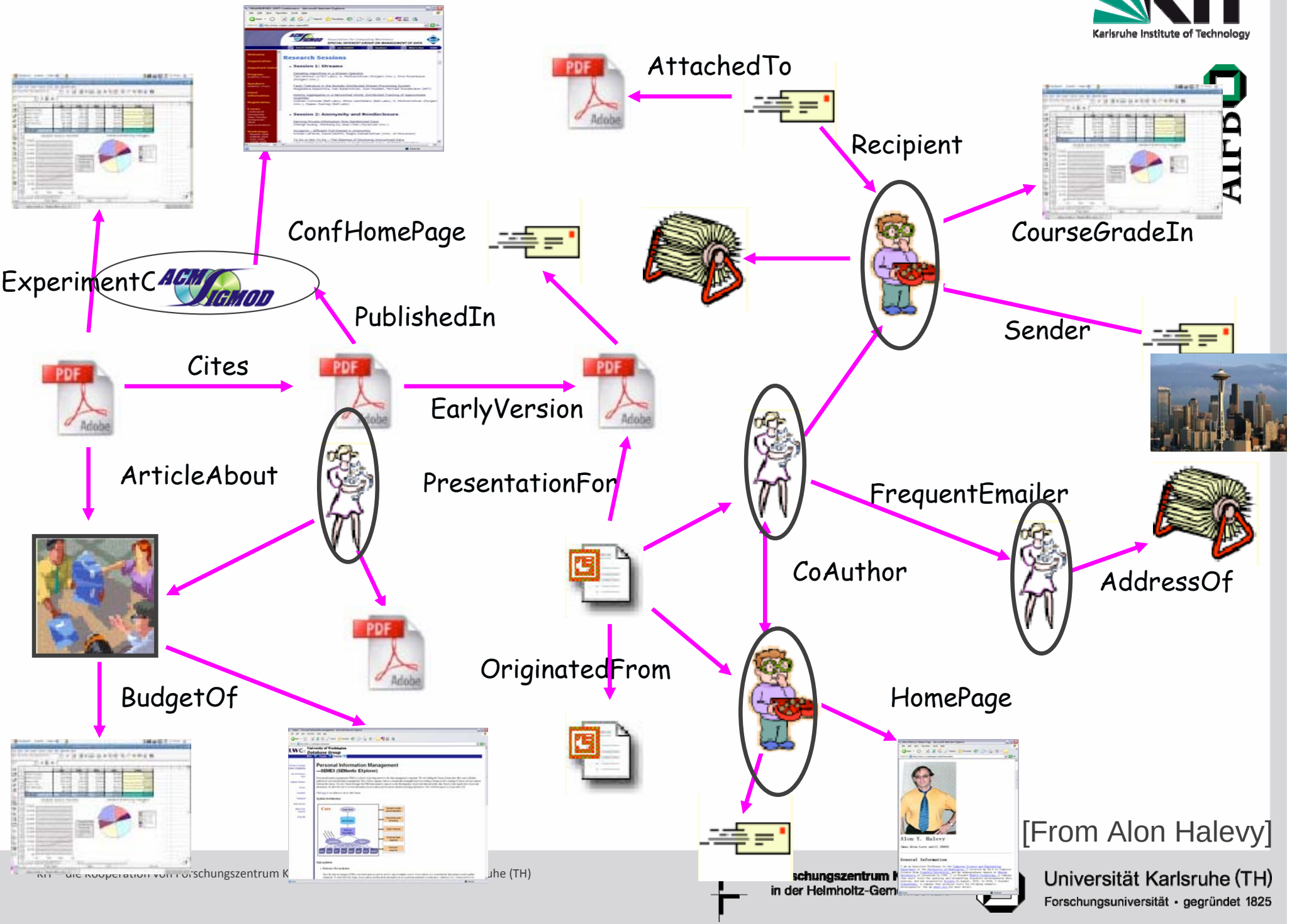


Why ontologies ?

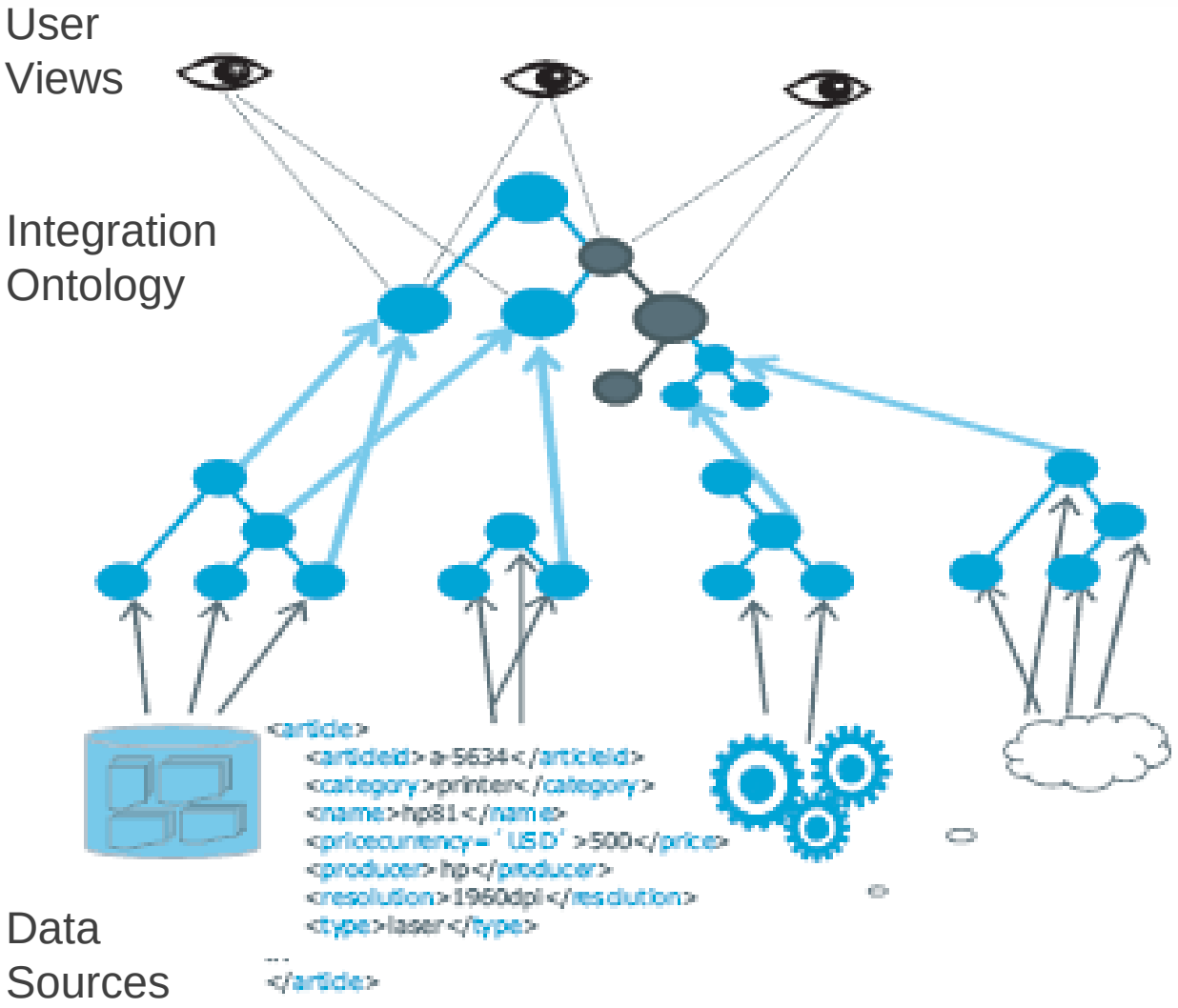
- **Flexibility** in capturing and integrating knowledge
- **Reusability** of knowledge across applications and domains

„Common technologies are not expressive enough to capture the complexity of relationships and adaptive enough to react to ever changing relationships. Semantic technologies make it possible in a flexible and maintainable fashion to capture all required complex relationships and make process them.“
(test manager at Audi)

Example: Personal Information Management



Example: Semantic Technologies for Enterprise Information Integration



- Data Integration **most costly** and challenging task for enterprises
- E.g. up to 80% of SAP migration costs due to data integration
- **Semantics** is the key issue in order to solve integration problems
- Semantic technologies enable dynamic integration solution which federate queries and merge data from heterogeneous data sources

Example: Enterprise Information Integration

- **Benefits** of ontologies
  - **Shared interpretation** of distributed data
  - **Conceptually adequate** and **expressive** data model to integrate heterogeneous data
  - **Ease of maintenance** because of declarative specification of integration rules

- **Reasoning services** are used for:
  - Computing relationships between information sources
  - Formal verification/validation of the integration result (e.g. inconsistency, redundancy, …)
  - Query answering across data sources

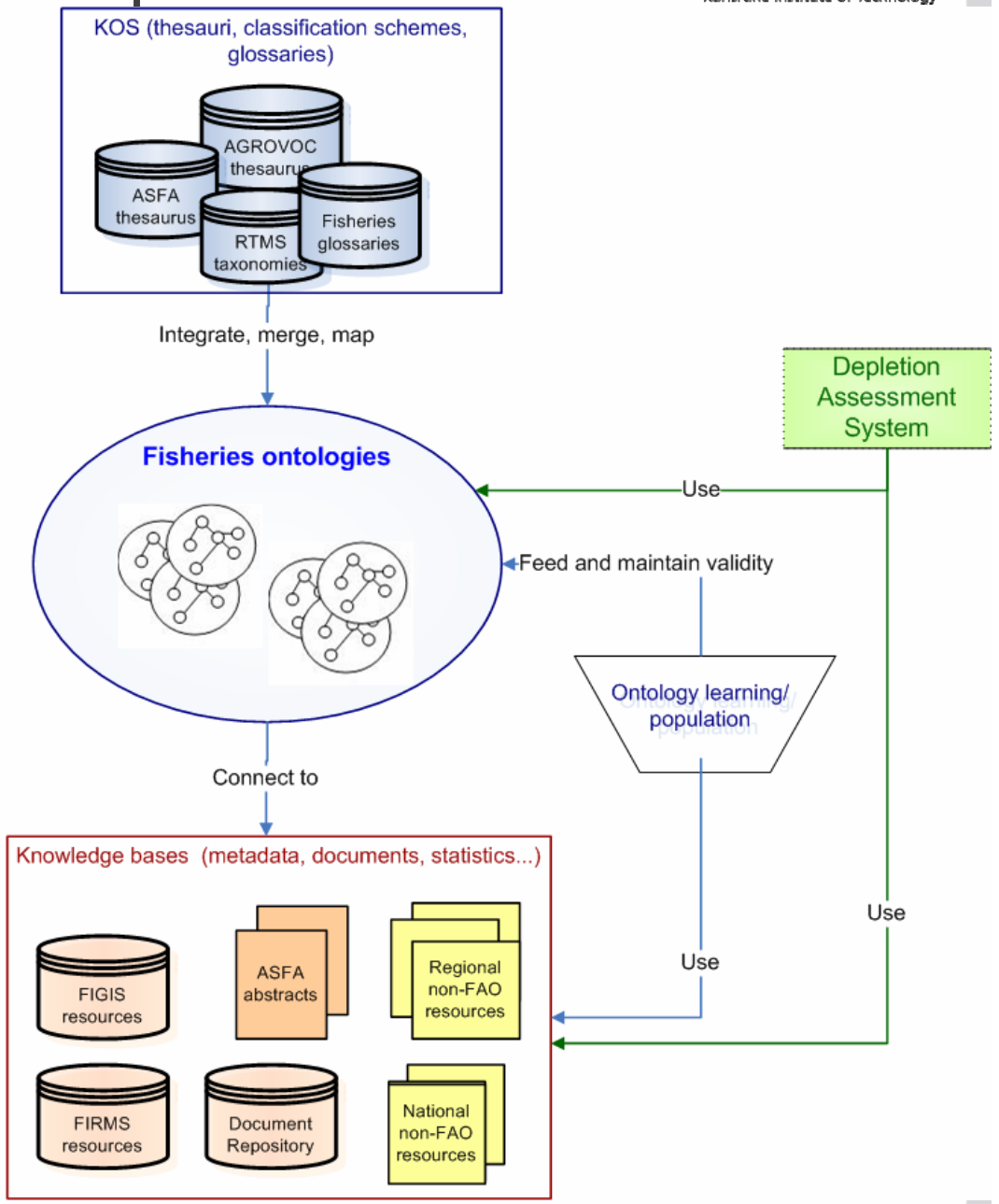
Example: FSDAS Application in the Fishery Domain

- FAO
  - **F**ood and **A**griculture **O**rganization of the United Nations
  - Case Study in the NeOn project

- **F**ish **S**tock **D**epletion **A**ssessment **S**ystem (FSDAS)
  - Bringing together related and relevant information
  - Discover and assess resources related to stock depletion
  - Decision support system to help Fisheries experts analyzing the status and trends of world's fish stocks

FSDAS – Ontology Runtime Requirements

- Ubiquitous and easy access to
  - status of fish stock
  - factors affecting fish depletion
- **Integration and querying** of heterogeneous (non-) ontological resources through the exploitation of the Fisheries ontologies
- return relevant results to the client
- Integrate with advanced annotation and visualization tools

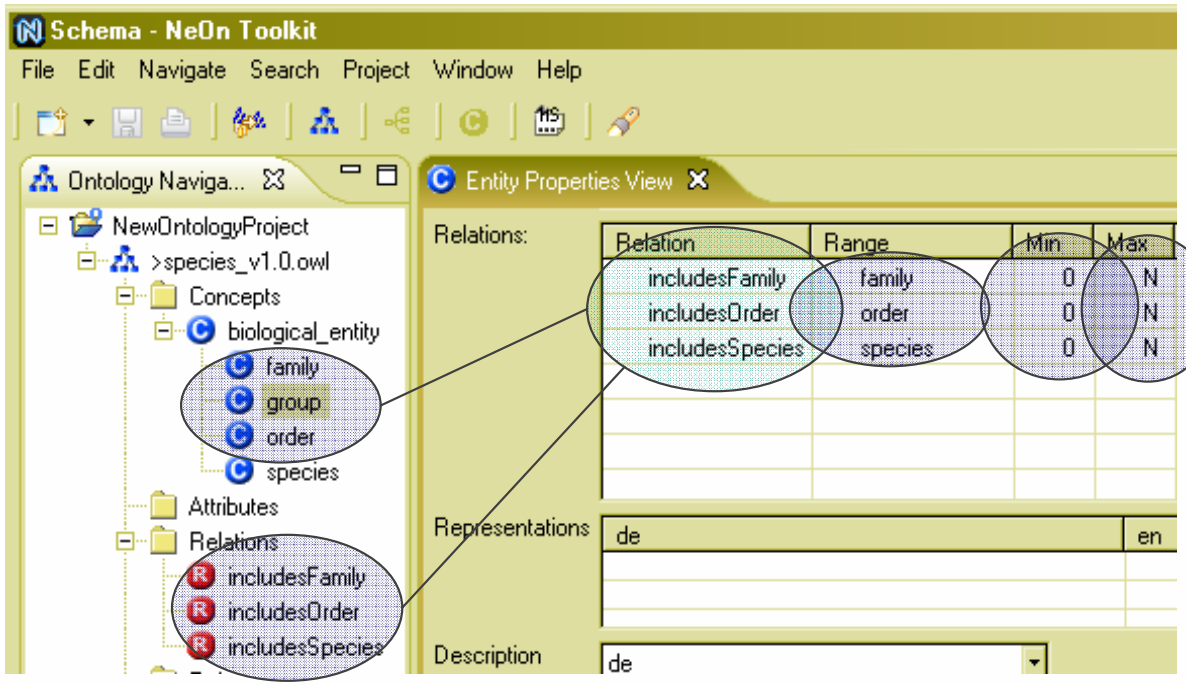


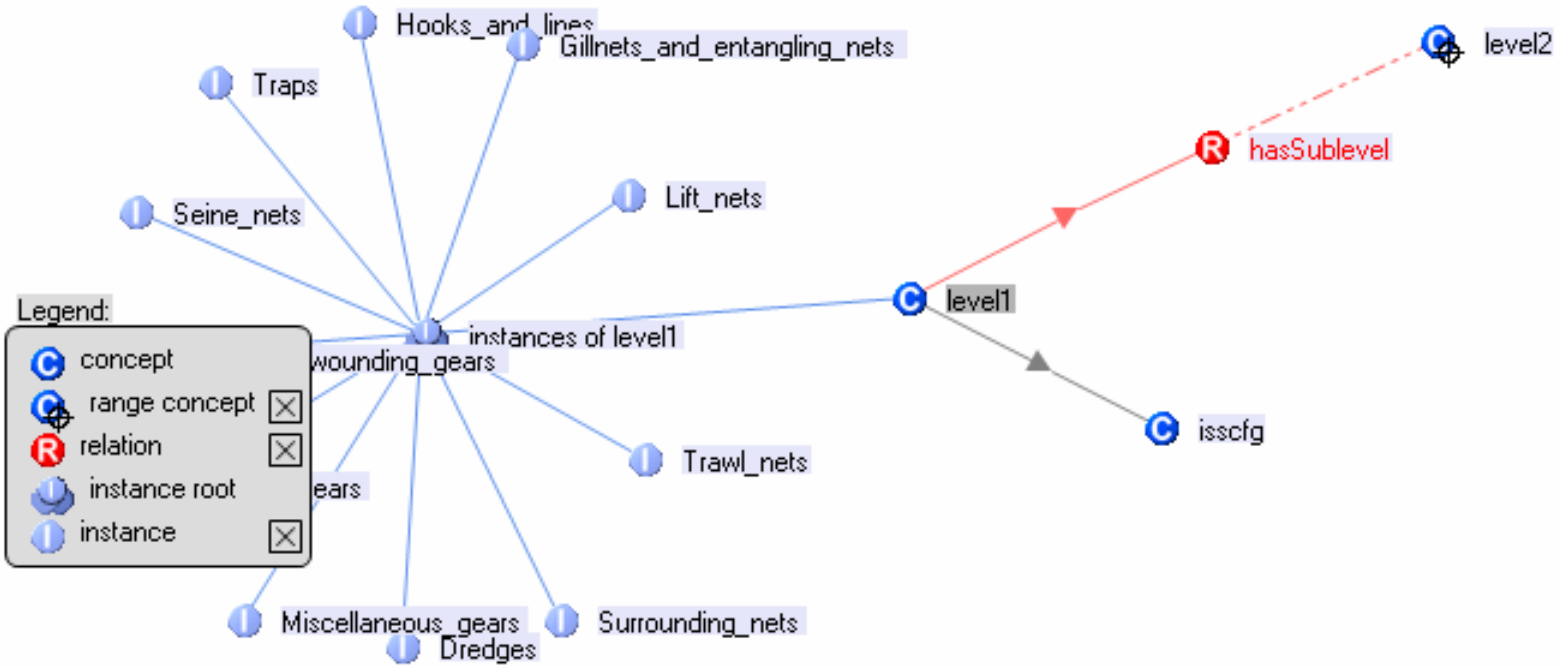
Fisheries Lifecycle Management – Ontology Engineering Requirements

- Provide support to ontology engineers and subject experts for:
  - modeling, populating, deploying, versioning ontologies
  - keeping them updated through an editorial workflow
  - managing mappings and relations between them

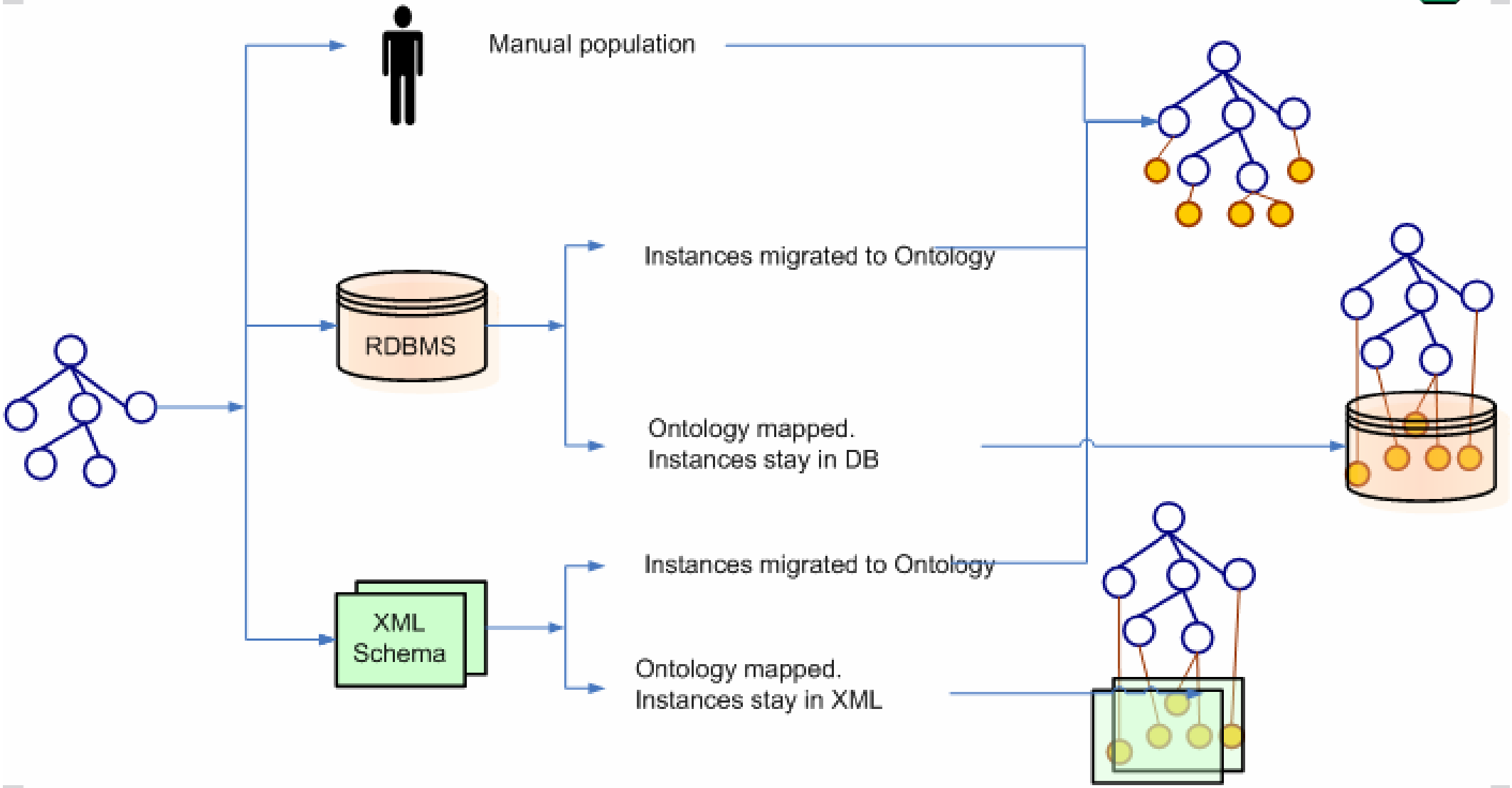
- Fisheries ontologies are:
  - multilingual ontologies
  - distributed / networked

Ontology Conceptualization

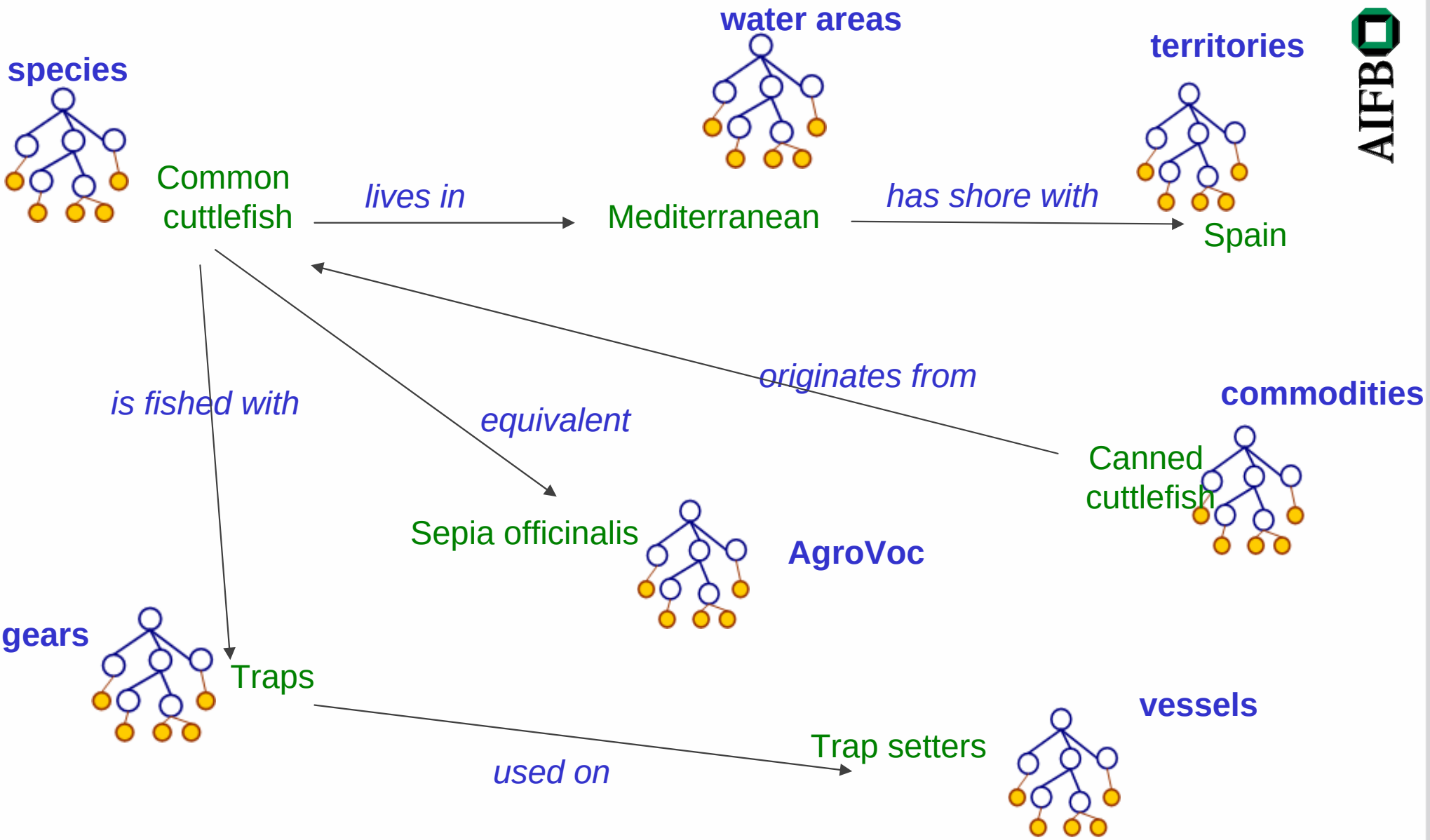




Ontology Population



Ontology Mapping to Create a Network



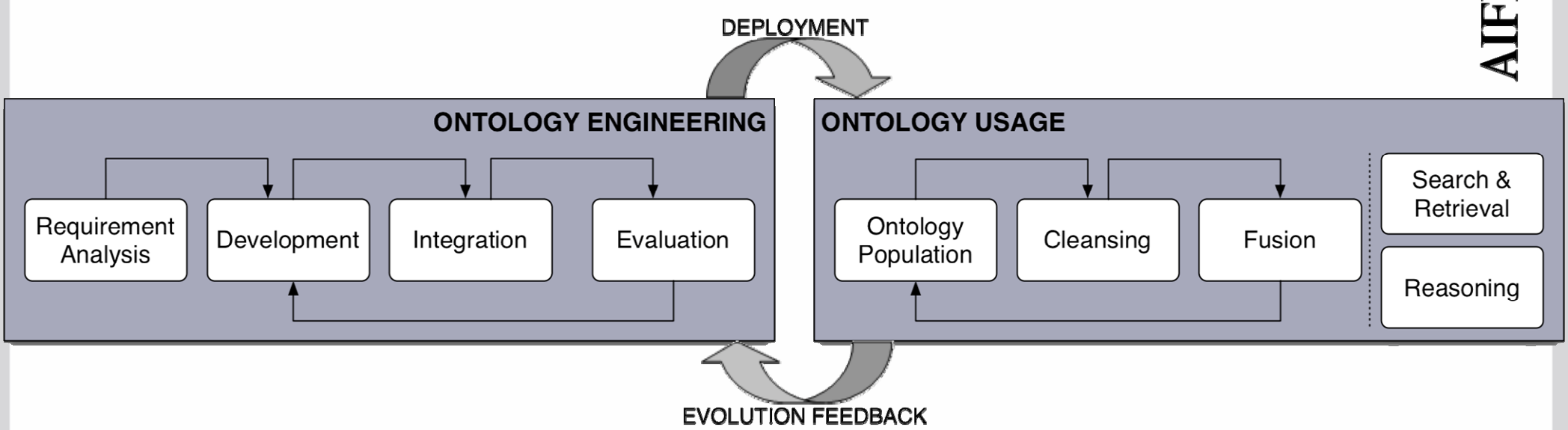
Aspects of Engineering Ontology–based Applications

- Engineering of ontology-based applications is a complex task, which involves
  - Ontology engineering
  - Software engineering
  - **Ontology management throughout the entire lifecycle**

➢ *Require systematic approach to*

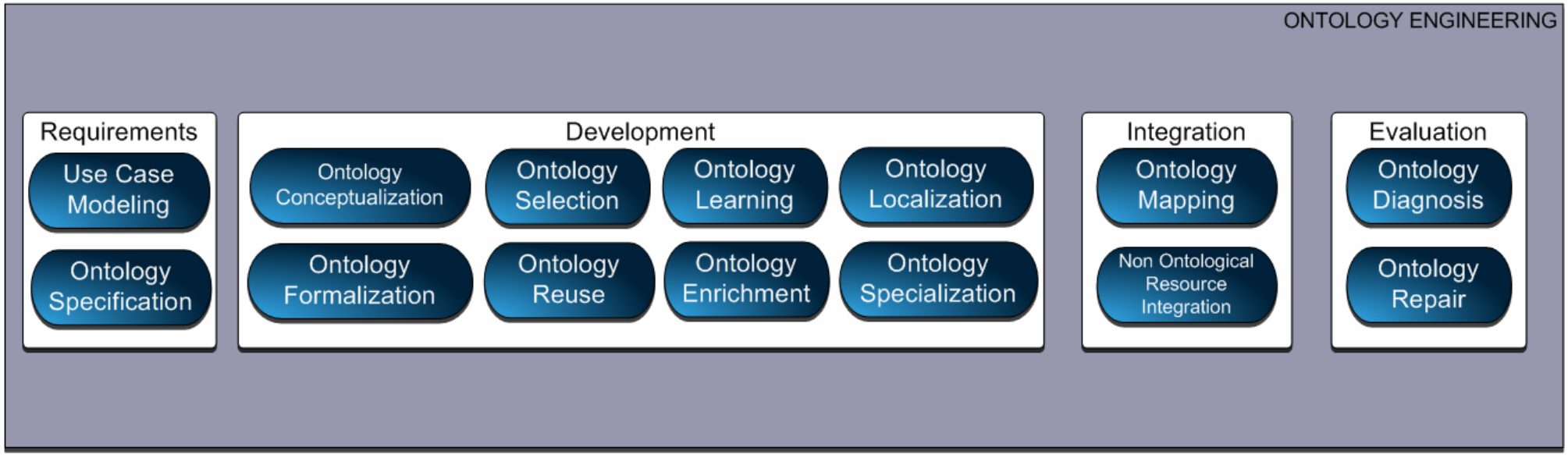
- the ontology and application development process,
- the ontology life cycle,
- the methods and methodologies for building ontologies and applications,
- and tool support

Ontology Lifecycle in Applications



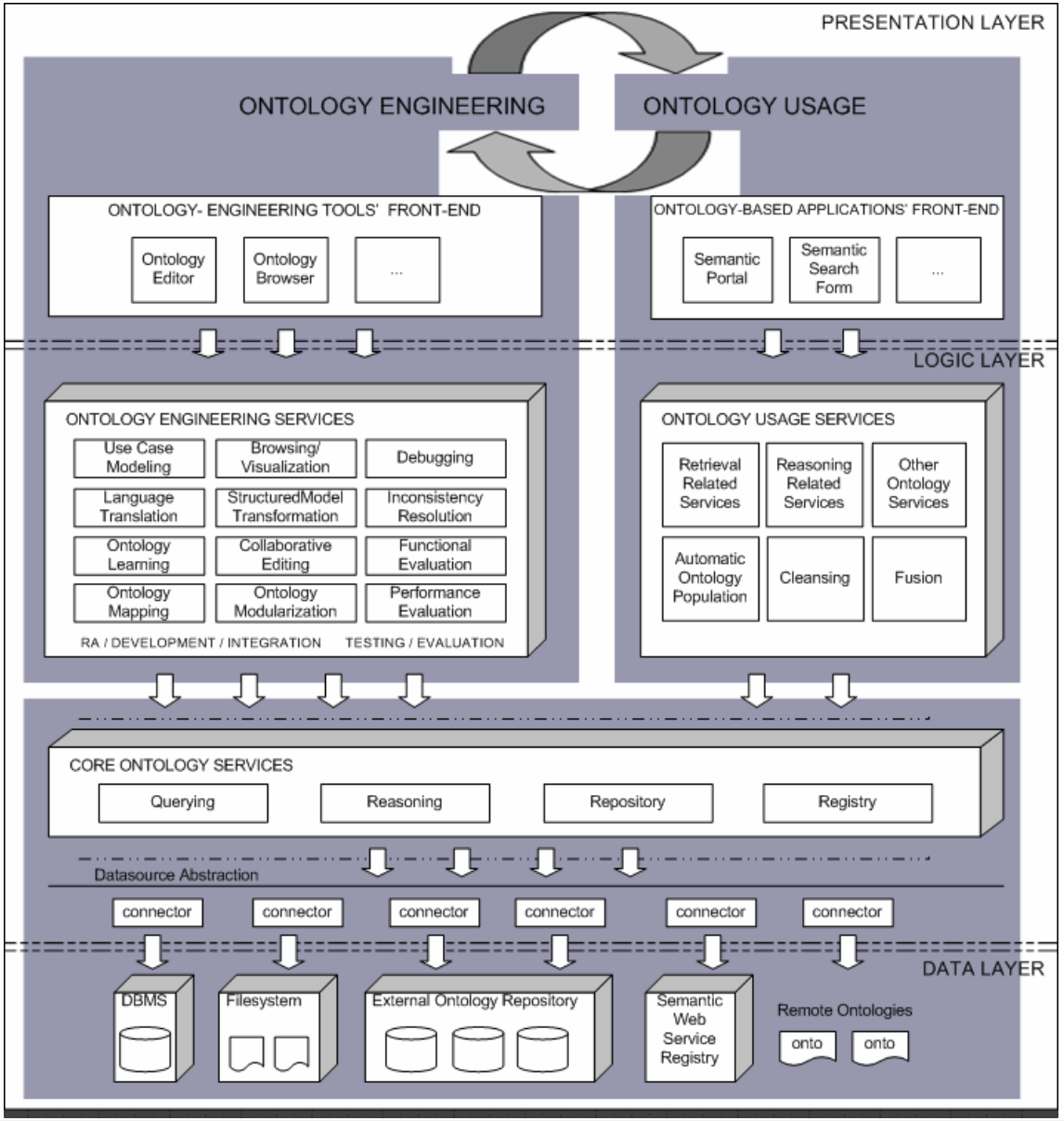
- Close interaction between ontology engineering and runtime usage

Ontology Engineering Activities



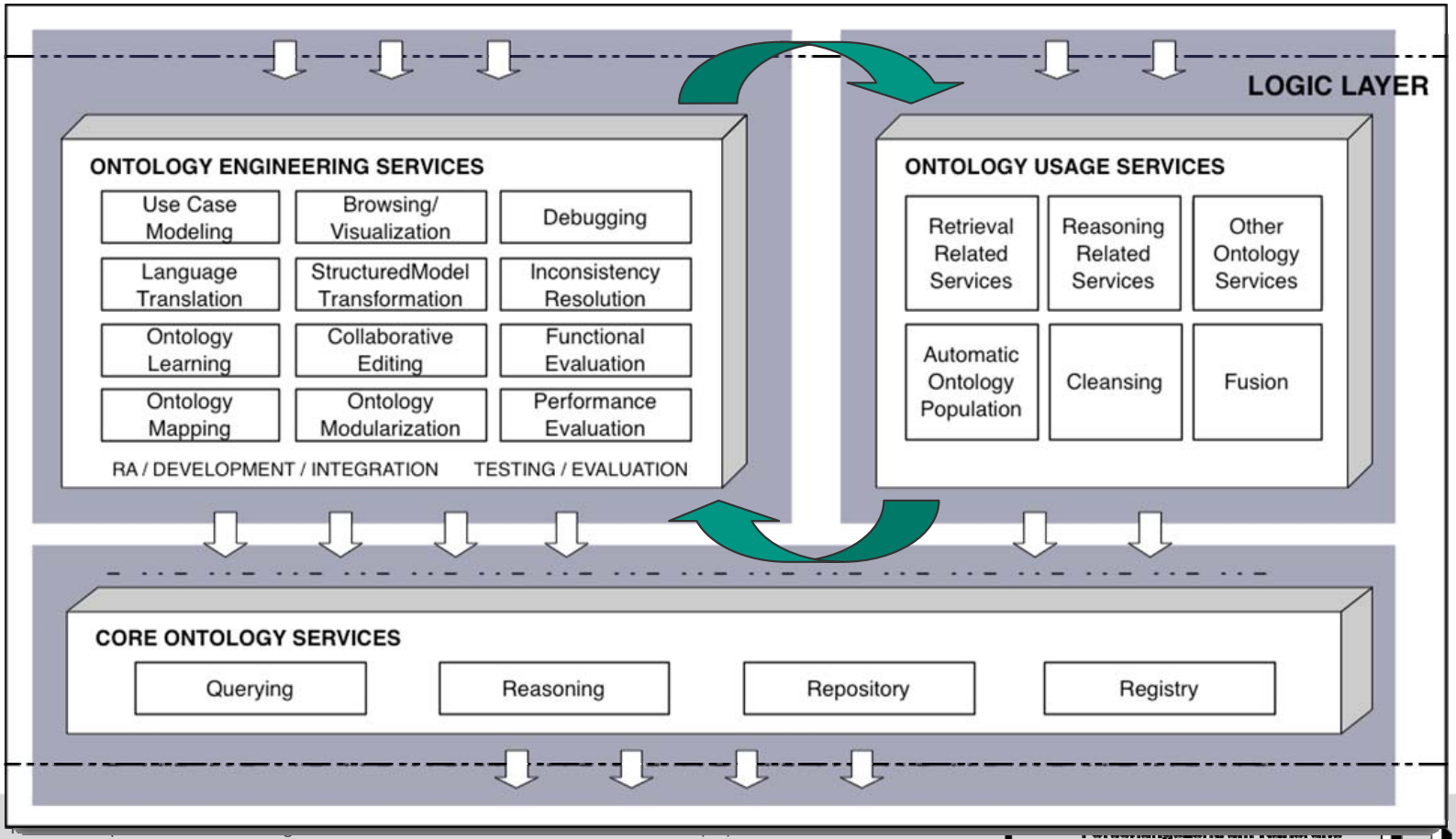
Generic Architecture with Lifecycle Support

- Inspired by SE best practices (SOA, JEE)
- Dynamic interaction of engineering and usage activities
- *Layered Organization*
- Presentation Layer
  - Thin client vs. Rich client
- Logic Layer
  - Business services / objects
  - Ontology services
- Data Layer
  - Ontological sources
  - Non-ontological sources



Architecture – Ontology Services

- Core services vs. higher level ontology lifecycle services
- Interactions within layers
  - May correspond to the structure of lifecycle activities, e.g. sequential flow
  - **Actual interaction depends on particular workflow of the use case**



Architecture – Core Ontology Services

- **Find and publish** ontologies
- Access, manipulation and storage of
  - Ontology
  - Ontology elements
- Querying, reasoning with ontologies

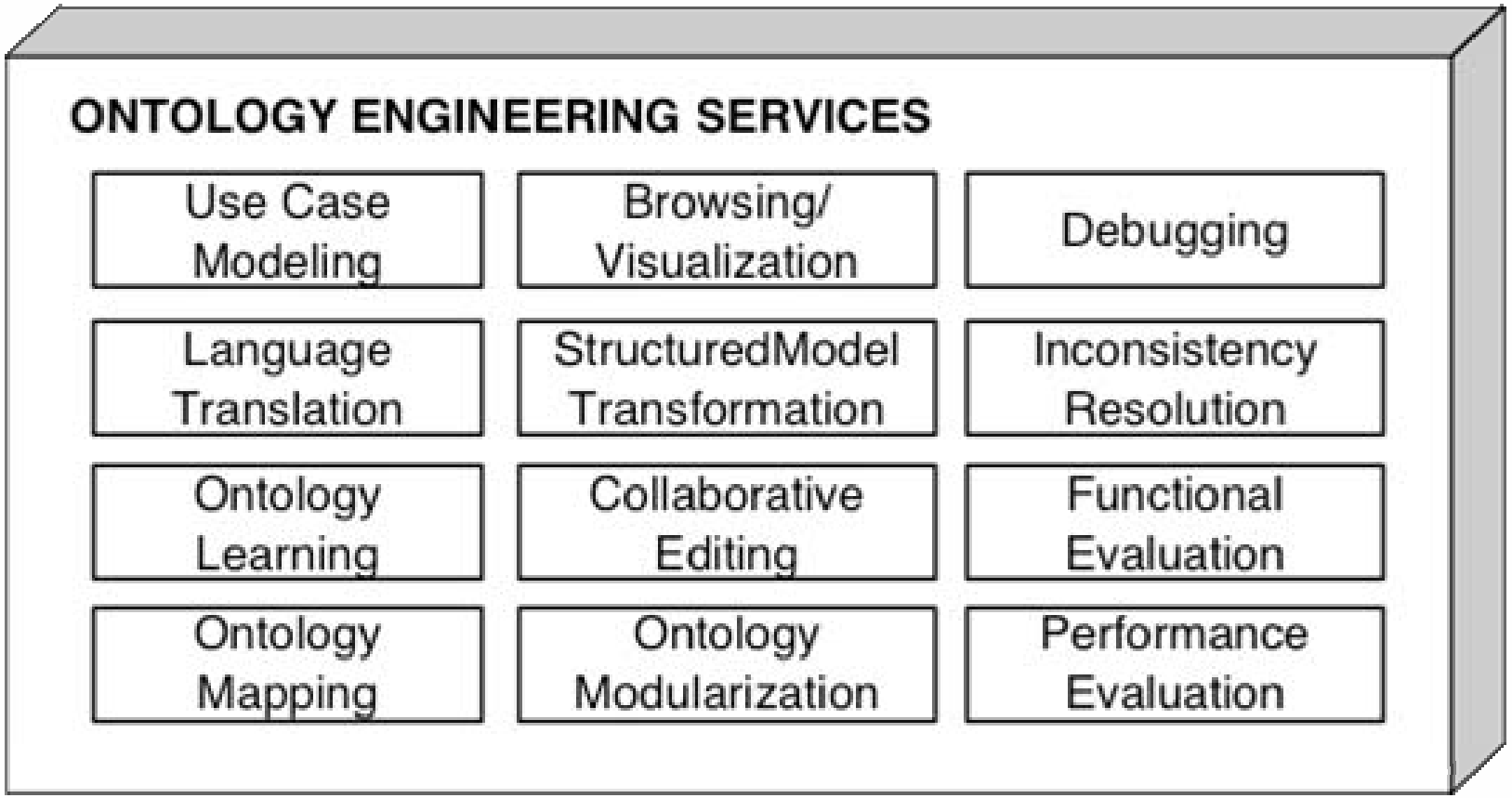
Core Ontology Services

- Registry
  - Registering and maintaining information *about* ontologies
  - Querying for available ontologies
  - Based on ontology metadata descriptions
- Repository
  - Focus on storing, manipulating and retrieving ontologies
  - File system vs. DBMS backend
- Querying
  - SPARQL query language
- Reasoning
  - Reasoning tasks (standard / non-standard)
  - DIG interface http://dig.sourceforge.net/

- *Often ontology management APIs cover many of the above at the same time*
  - Jena API, KAON2 API, OWL API

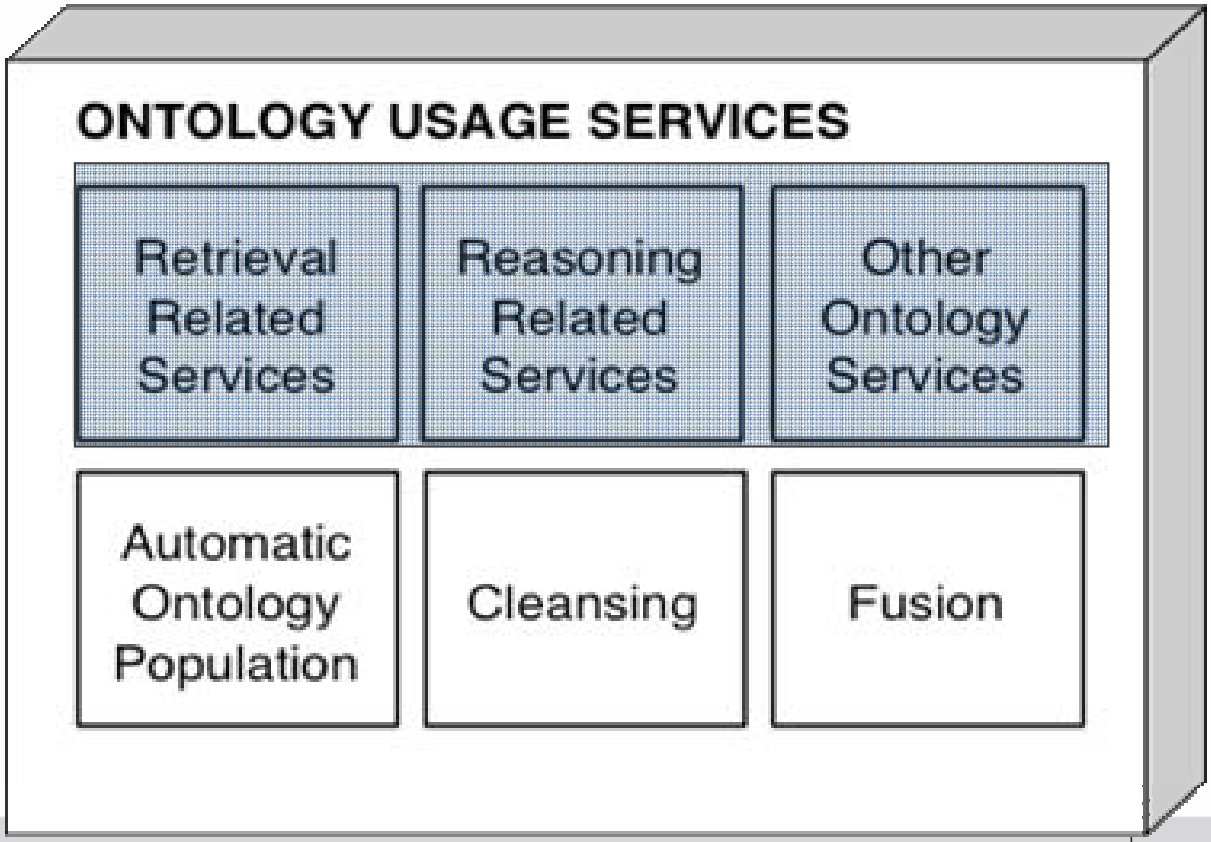
Architecture – Ontology Engineering Services

- Services for **engineering activities**
  - Requirement analysis, Ontology Development, Ontology Integration, Ontology Evaluation



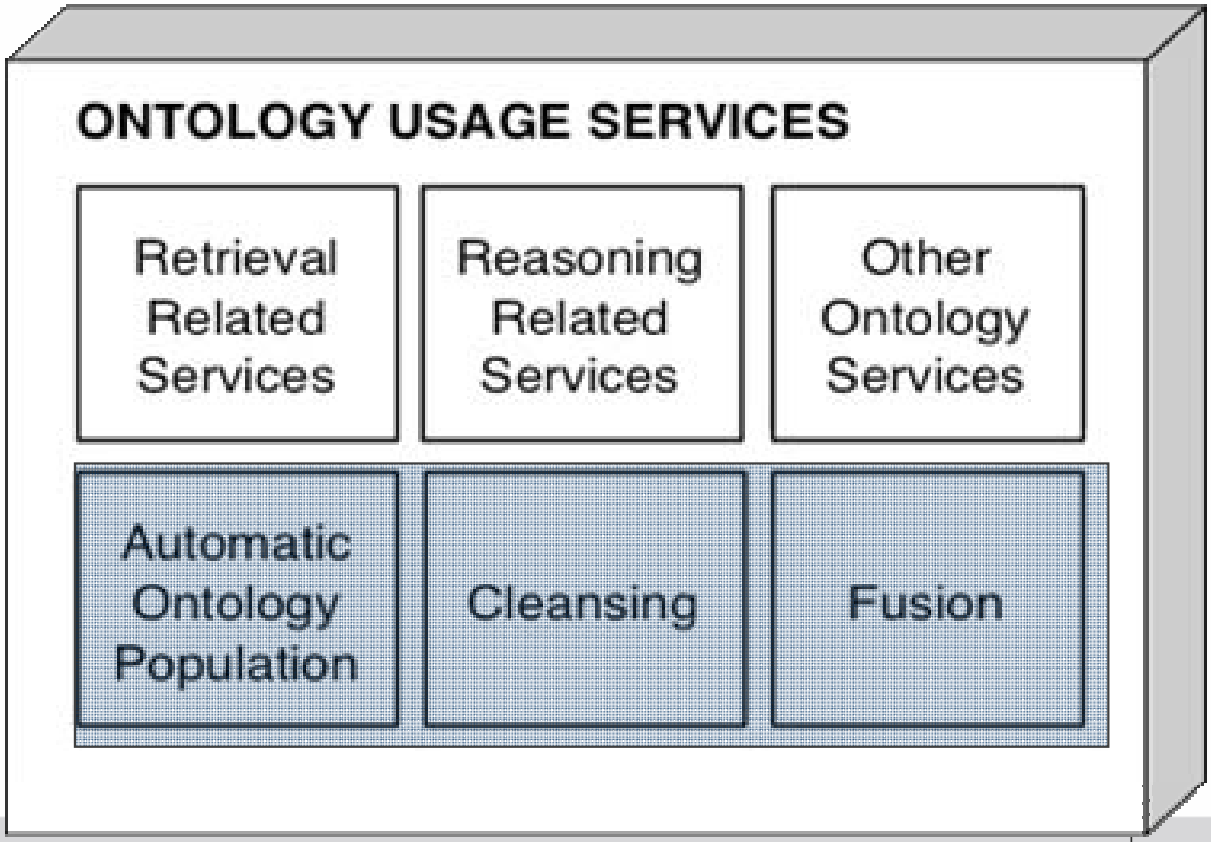
Architecture – Ontology Usage Services

- **Application-specific usage services**
  - Involve reasoning, retrieval and other tasks enabled by ontologies
  - Implement use cases



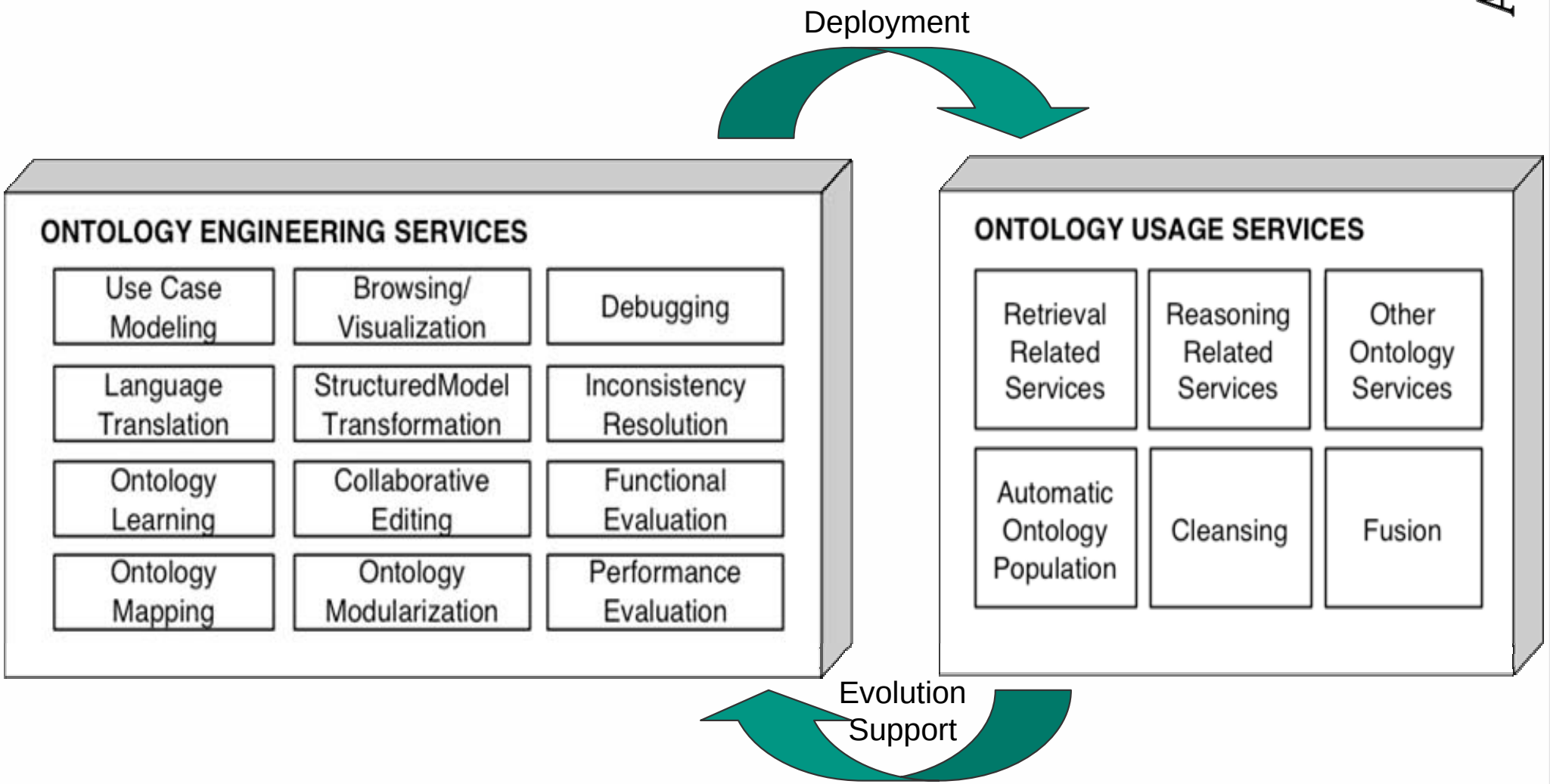
Architecture – Ontology Usage Services

- **Lifecycle usage services**
  - Population, Cleansing, Fusion



Architecture – Ontology Services

- **Feedback Loop**

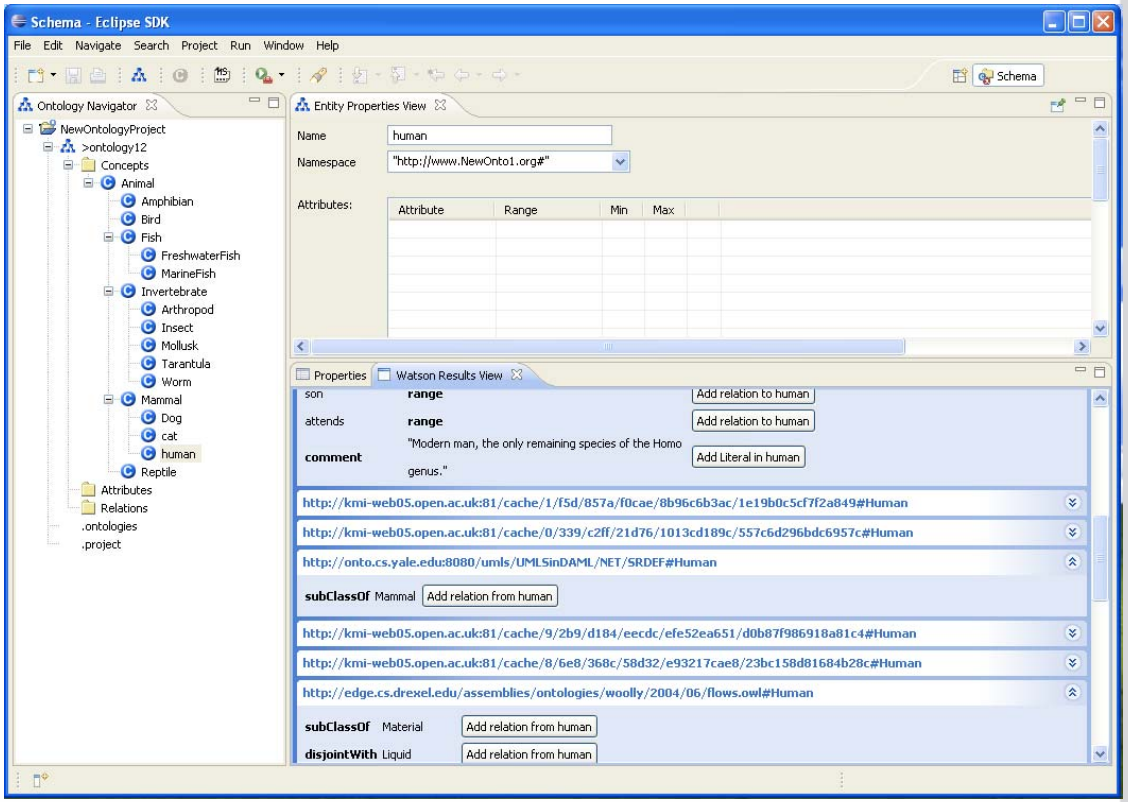


Application Design and Implementation

- **Design and implementation of concrete applications based on generic architecture**
  - **Match technical requirements**
    - Degree of distribution, coupling and granularity of components
    - Choose **concrete architecture paradigm** (JEE, Rich Client Application, …)
  - **Match functional requirements** against generic services and components

NeOn Toolkit

- Reference implementation of the generic architecture
- Infrastructure and reusable software components
- Based on Eclipse platform
- Based on a set of standardized APIs
- Plugins implement ontology services

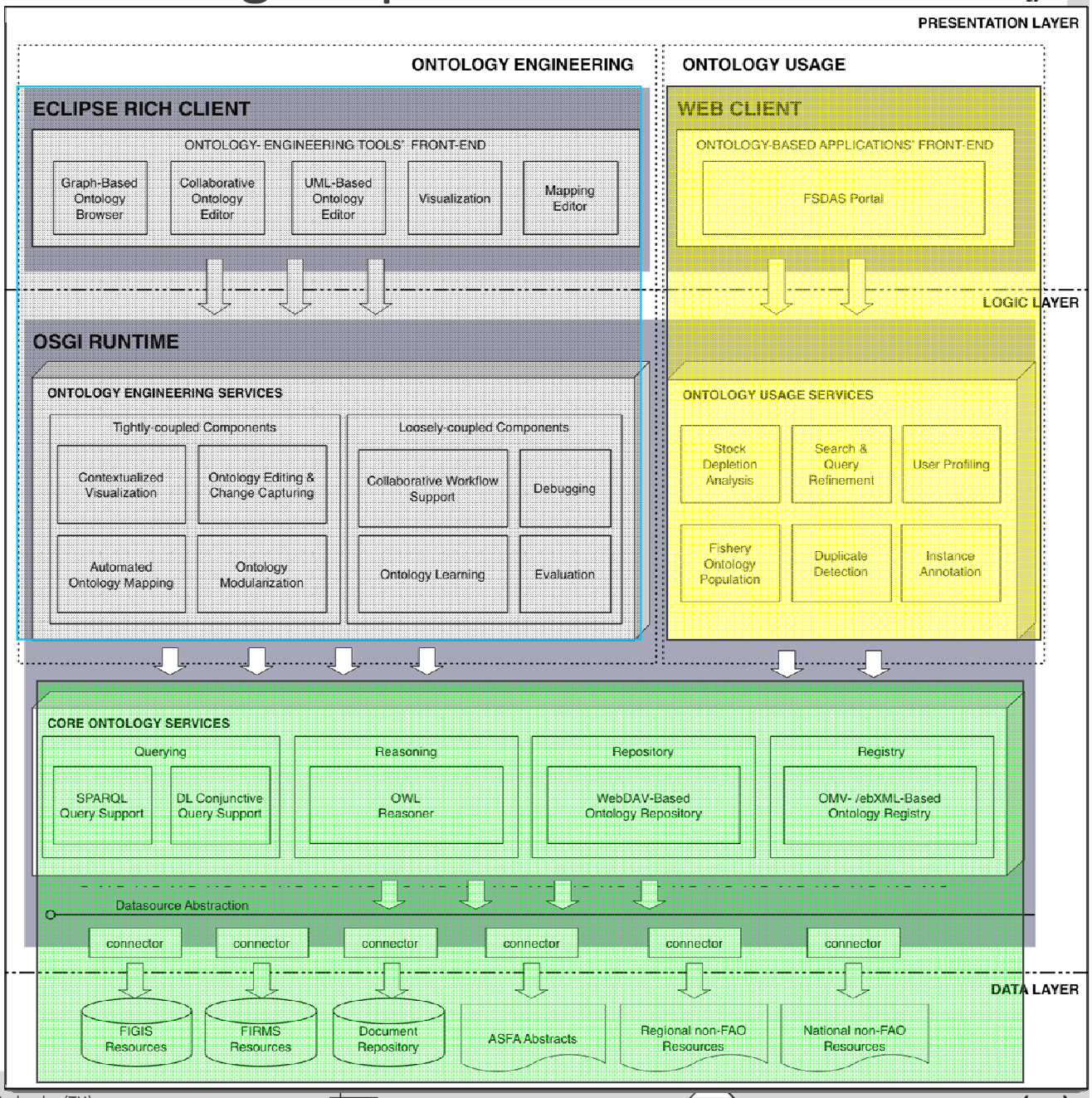


http://www.neon-toolkit.org/

➢ *Exercise on 28.4.2008*

FSDAS Application – Matching Requirements

- Web-based application
  - For end users
  - Ubiquitous access
- Rich client application
  - For ontology engineers
  - Rich set of tools
- Realized as bundle of
  - NeOn Lifecycle services
  - Loosely coupled services
  - Tightly coupled components
- **NeOn backend infrastructure**
  - Based on OSGi / Eclipse
  - Core ontology services



Summary

- Wide spectrum of types of ontologies
- Wide spectrum of ontology-based applications
- Generic Architecture
  - Guidance for the **design of ontology-based applications**
  - For use cases where ontology usage and engineering are intertwined at runtime → dynamic feedback loop
- NeOn Toolkit
  - Reference implementation of generic architecture
  - **Reusable lifecycle components**