

Intelligent Internet Technologies



Lecture 24.

Inference in Ontologies (Reasoning). Applications and Tools

Alexandra V. Vitko

KNURE, AI Department, alexandra_vitko@yahoo.com



Why Reasoning ?



- **Ontology design**
 - Check class consistency and (unexpected) implied relationships
 - Particularly important with large ontologies/multiple authors
- **Ontology integration**
 - Assert inter-ontology relationships
 - Reasoner computes integrated class hierarchy/consistency
- **Ontology deployment**
 - Determine if set of facts are consistent with respect to ontology
 - Determine if individuals are instances of ontology classes
 - Query Inclusion
 - Service description matchmaking
 - Classification-based querying.

Reasoning Types



- Tool support for three types of reasoning exists:
 - **Consistency checking:**
Can a class have any instances?
 - **Classification:**
Is A a subclass of B? Are two classes equivalent?
 - **Instance classification (Instantiation):**
Which classes does an individual belong to?

Simple Inference: Instance Classification

- Given the definition for the property **hasParent** with domain=**Animal**, range=**Animal** and

```
<owl:Thing rdf:ID="Tuzik">  
  <hasParent rdf:resource="#Layka"/>  
</owl:Thing>
```

we can infer that **Tuzik** is an **Animal** and **Layka** is an **Animal**

Classification



NationalPark

Asserted Conditions

NECESSARY & SUFFICIENT

NECESSARY

RuralArea

\exists hasAccommodation Campground

\exists hasActivity Hiking

BackpackersDestination

Asserted Conditions

NECESSARY & SUFFICIENT

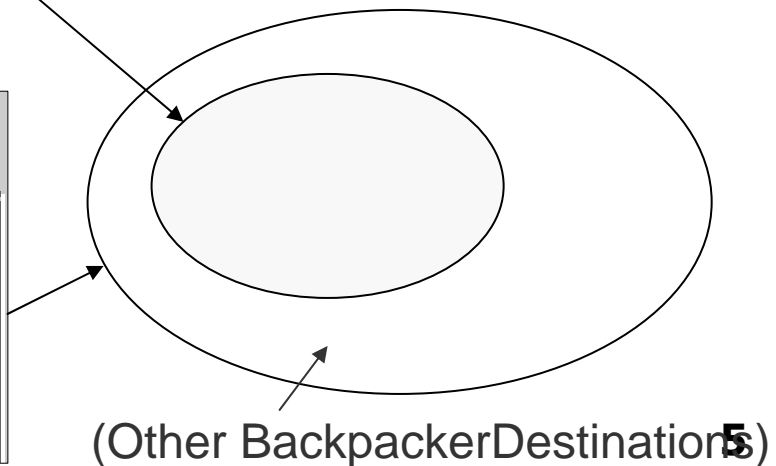
NECESSARY

Destination

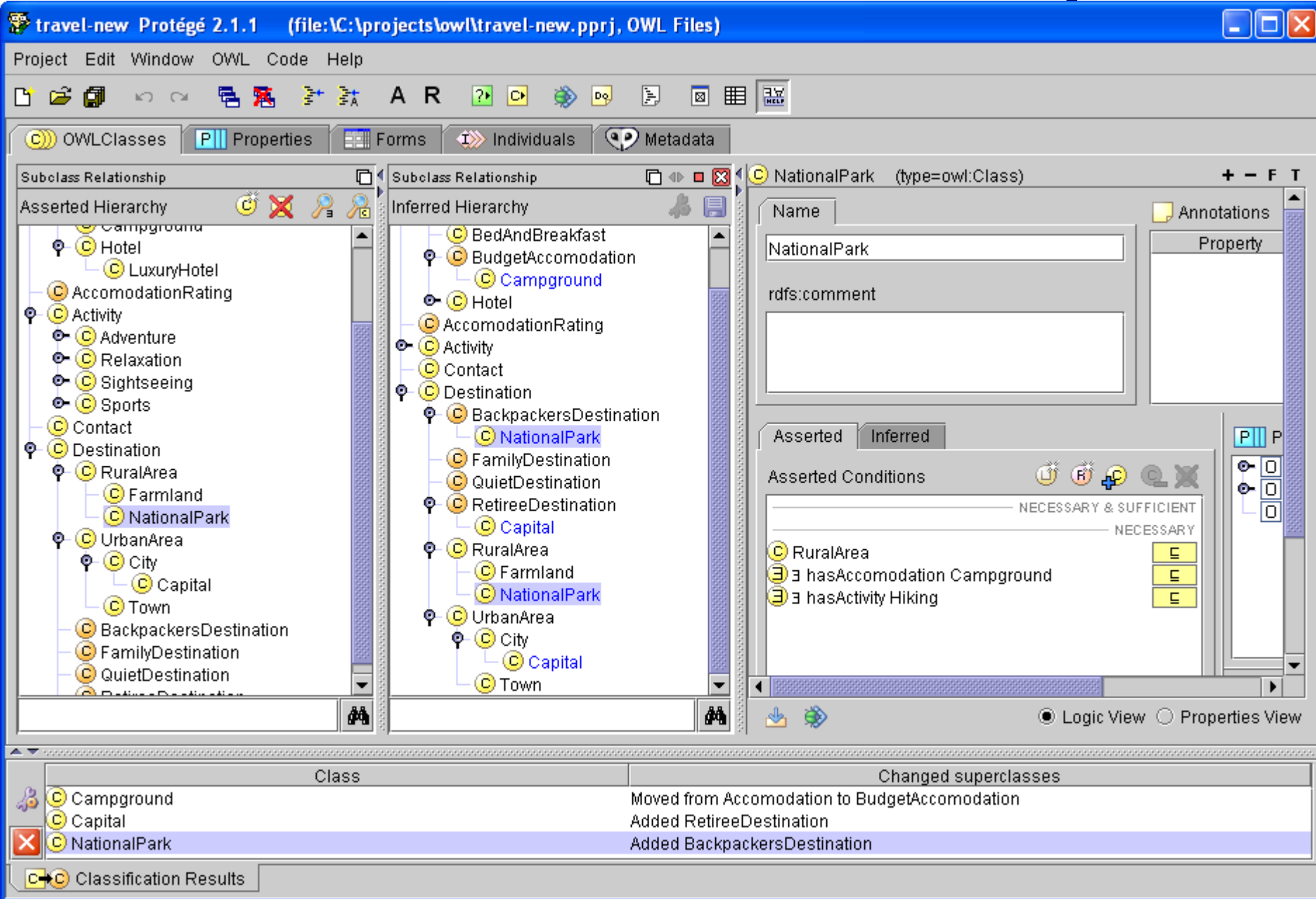
\exists hasAccommodation BudgetAccommodation

\exists hasActivity (Sports \sqcup Adventure)

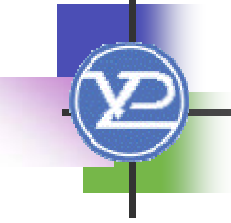
- A RuralArea is a Destination
- A Campground is BudgetAccommodation
- Hiking is a Sport
- Inference:
Every NationalPark is a Backpackers-Destiantion



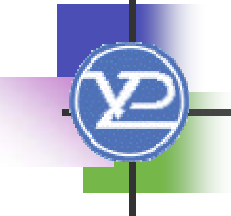
Classification results in Protege



Inference in Protege

- 
- Protégé OWL plug-in
 - Reasoner “Racer” for inference
 - Description Logic based reasoning engine
 - Server-based
 - Integrates with Protégé-OWL

Need for Inference Tools

- 
- OWL is about content, not the syntax
 - Statements from different documents about the same URI are automatically conjoined
 - OWL can appear unintuitive to the uninitiated
 - Declare that no one can have more than one mother
 - Declare Mary is John's mother
 - Declare Jane is John's mother
 - An OWL reasoner would say $Mary = Jane$

OWL Inference Tools



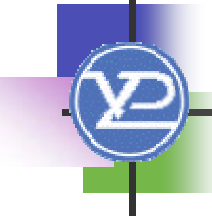
- **Racer** (see above)
- **Pellet**
 - DL based reasoner implemented in Java
- **Euler**
 - an inference engine supporting logic based proofs. Finds out whether a given set of facts support a given conclusion
- **FaCT**
 - DL classifier

OWL Tools (2)

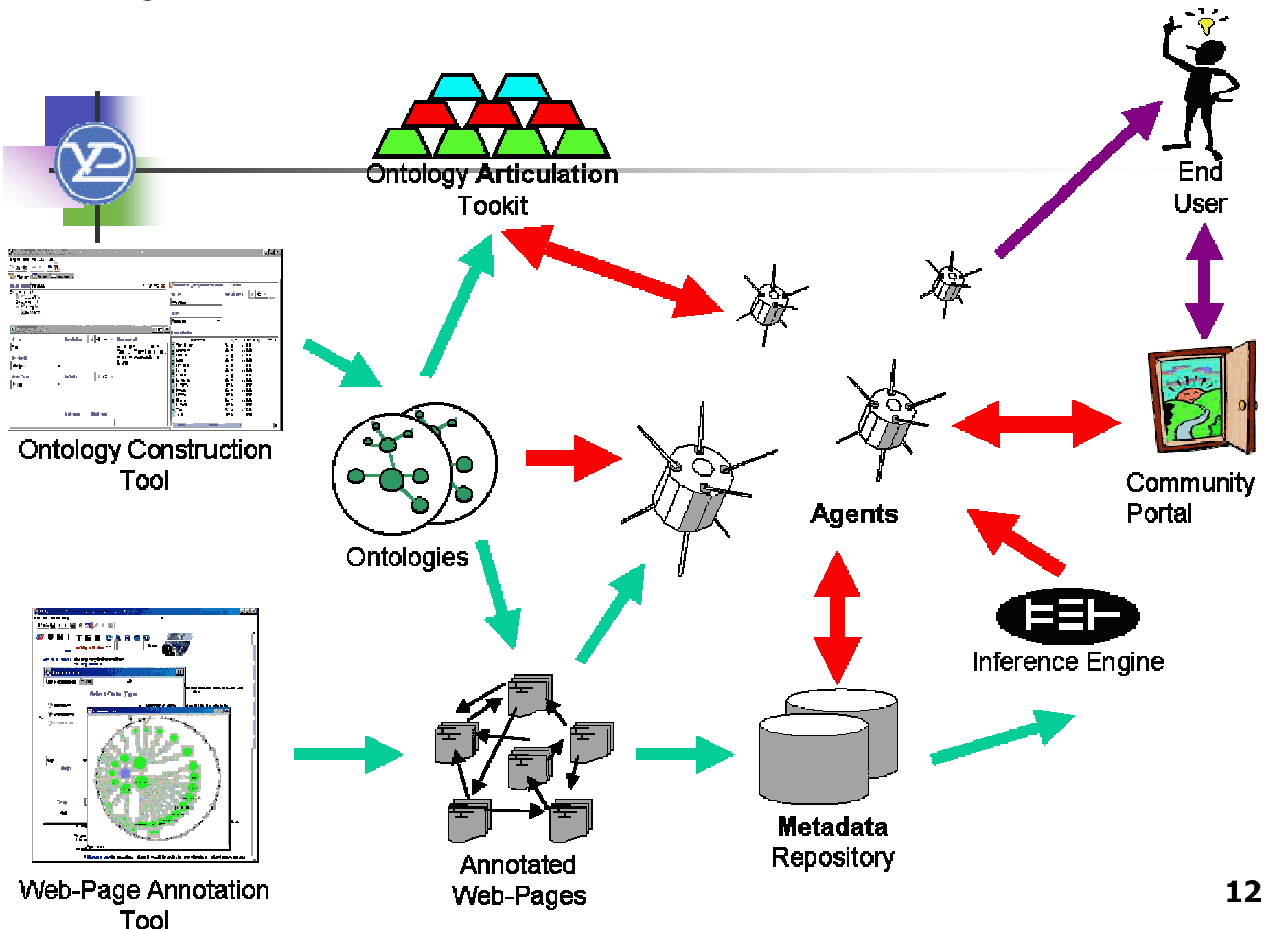


- Hoolet
 - DL Reasoner that uses a First Order Prover to reason about ontologies
- Jena2
 - sound (but not complete) instance reasoning for OWL Lite
- Surnia
 - OWL Full reasoner
- etc.

Other Types of Ontology Tools

- 
- Being able to express ontologies is not enough...we need tools!
 - Existing tools, especially for DAML+OIL, are adapting to OWL
 - Types of tools:
 - Ontology construction tools
 - Annotation tools
 - Inference engines (Reasoners)
 - Ontology articulation (integration) tools

Reasoning



Ontologies in Applications

- In the past, ontologies have been **embedded** in the application
- This introduces problems with:
 - Maintenance
 - Implementation
 - Exchange of terms
- Considering the ontology as a separate resource provides greater flexibility

Implementations of Ontologies



- What does it mean to provide implementations of ontologies?
- What do you want to **do** with the ontology?
 - Read it
 - Manipulate it
 - Reason about it
 - Where does the reasoner go?
 - Inside or outside?

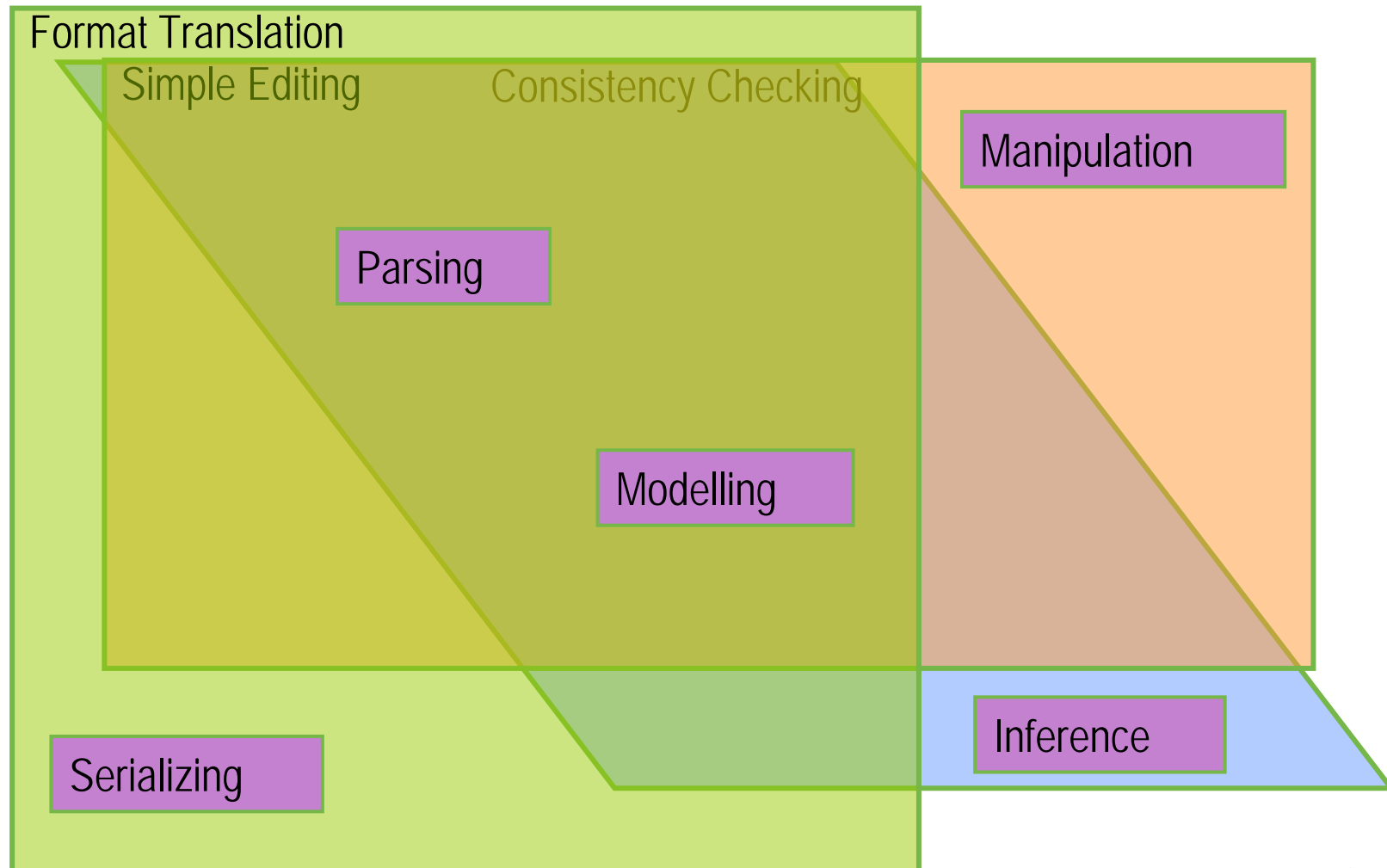
Ontology Interactions



- **Modelling**
 - Provide data structures that represent OWL ontologies/documents
- **Parsing**
 - Taking some syntactic presentation, e.g. OW-RDF and converting it to some [useful] internal data structure
- **Serialization**
 - Producing a syntactic presentation, e.g. OWL-XML from a local data structure
- **Manipulation**
 - Being able to manipulate the underlying objects
- **Inference**



Implementation Aspects

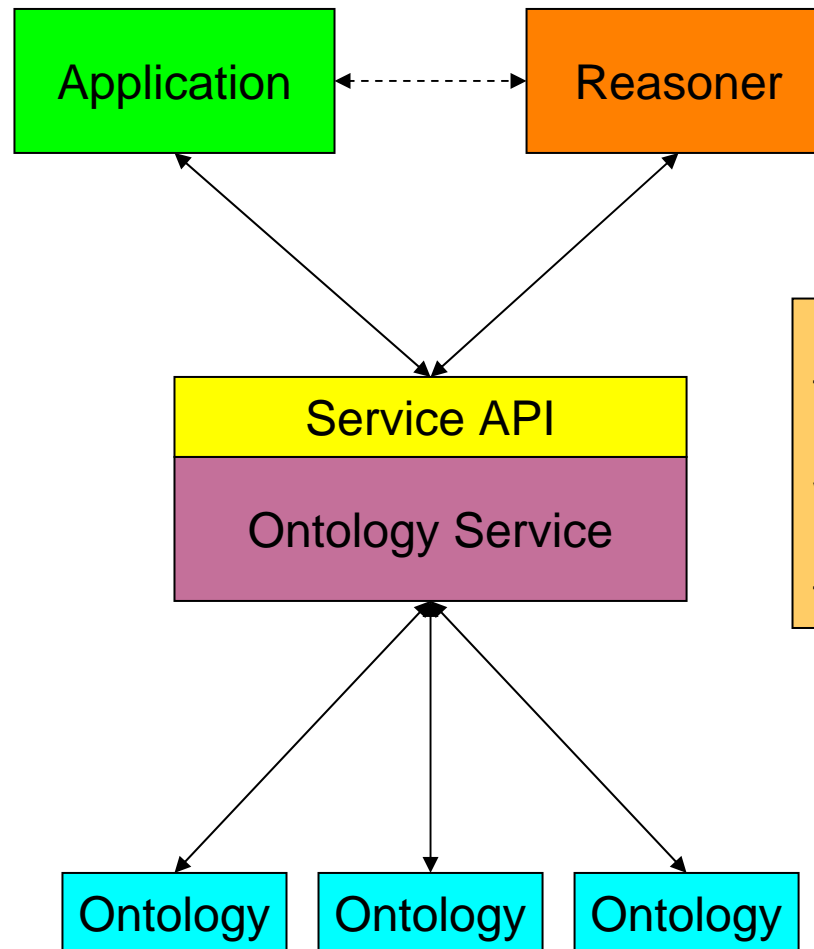


Layering



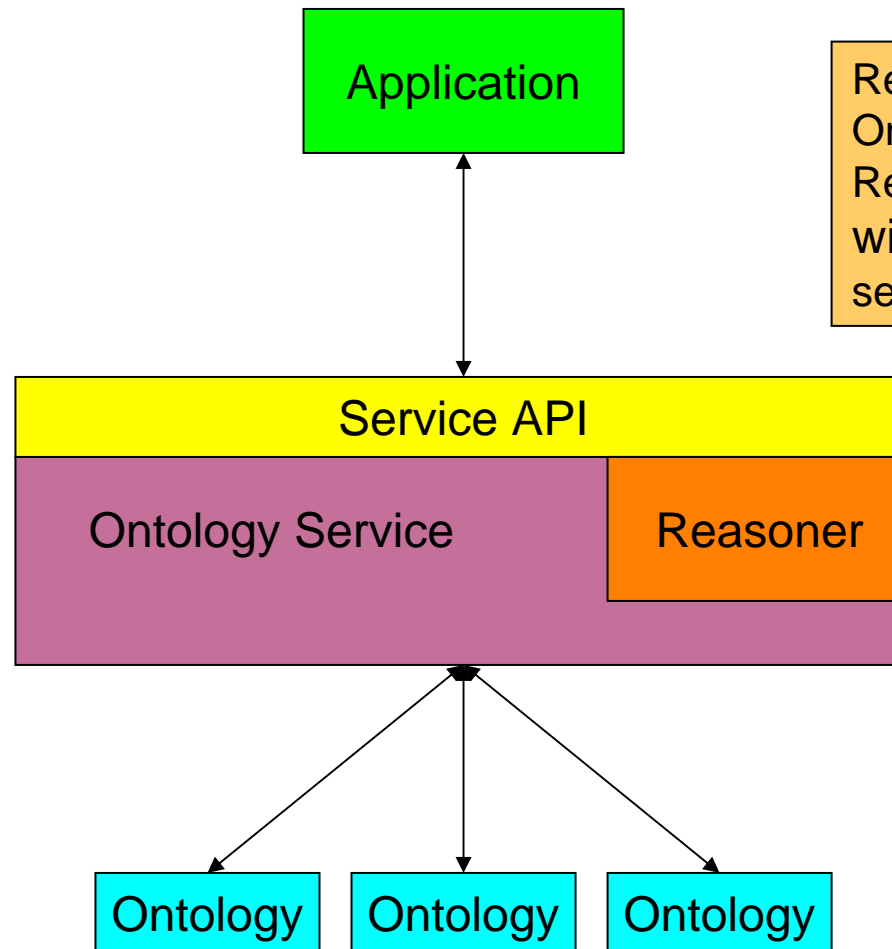
- OWL is layered on RDF.
- This layering provides us with a number of different options in terms of accessing the ontology.
- Direct access to the RDF triple structures
 - E.g. Jena, Sesame, 3store
 - May require client applications to “understand” or implement aspects of the language, e.g. inference.
- Access at a “higher level” using some API.
 - OWL-API, Protégé API, Jena Ontology API

Reasoner Outside



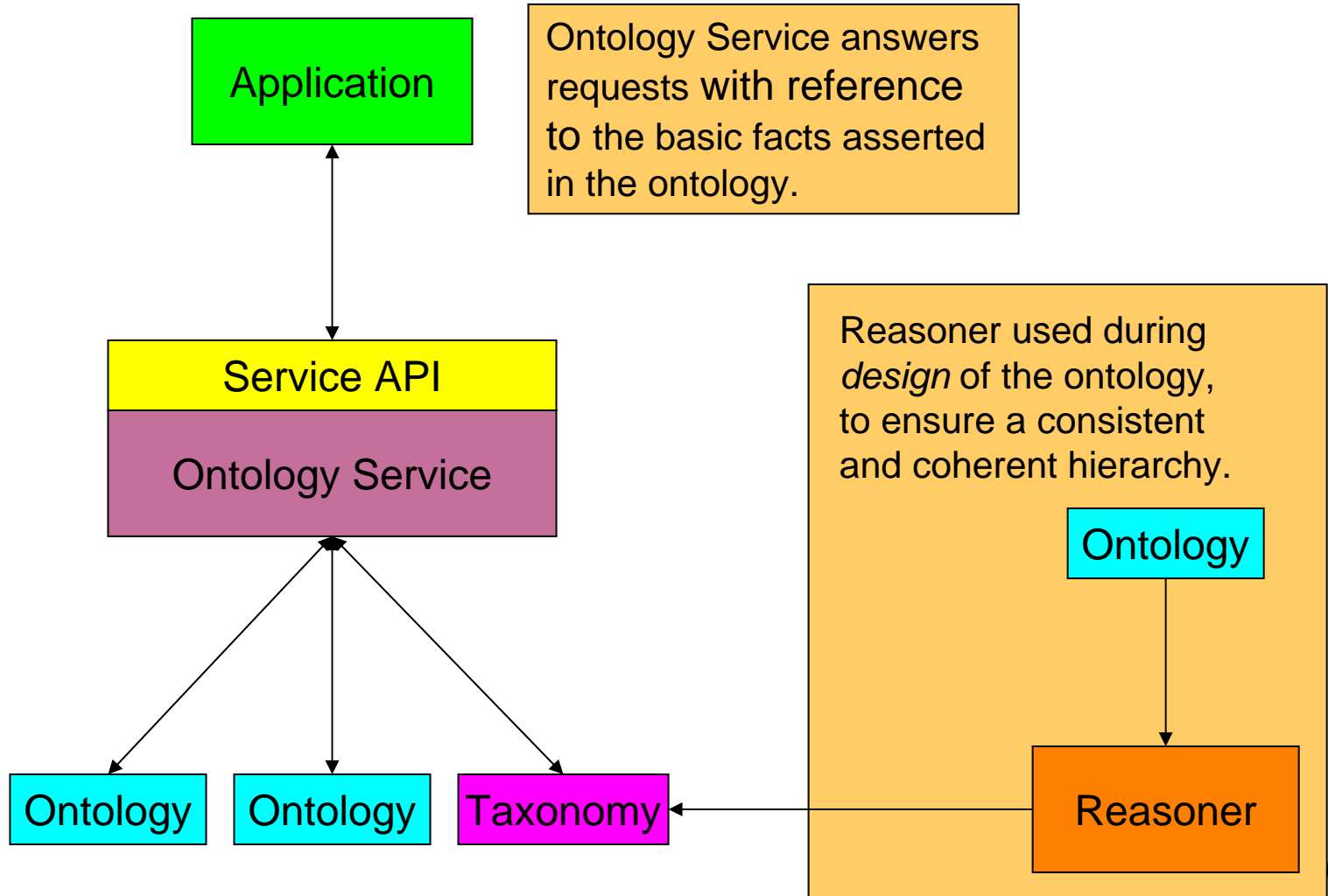
Reasoner sits outside the Ontology Service. Requests are answered with reference to the basic facts asserted in the ontology.

Reasoner Inside

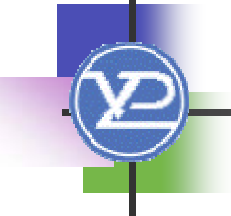


Reasoner sits inside the Ontology Service. Requests are answered with reference to the semantics.

Do You Really Need Reasoning?



Summary

- 
- Reasoning services help knowledge engineers to check consistency of ontologies, to make them more full
 - Reasoning services help user to query ontologies
 - Ontology tools are increasingly available

МУ к курсовому проекту



- ftp://10.12.57.254/upload/vitko/ITI/Course Project/IIT_course-project.pdf
- Проверить содержание пояснительной записки и требования
- Проверить ГОСТы !!!