Hackathon Project (HC-07): Ontohub-OOR-OOPS! Integration

Till Mossakowski[1,2]
Ontology Summit 2013

[1]SFB/TR 8 "Spatial cognition", University of Bremen, Germany
[2]DFKI GmbH, Bremen, Germany

2013-05-03

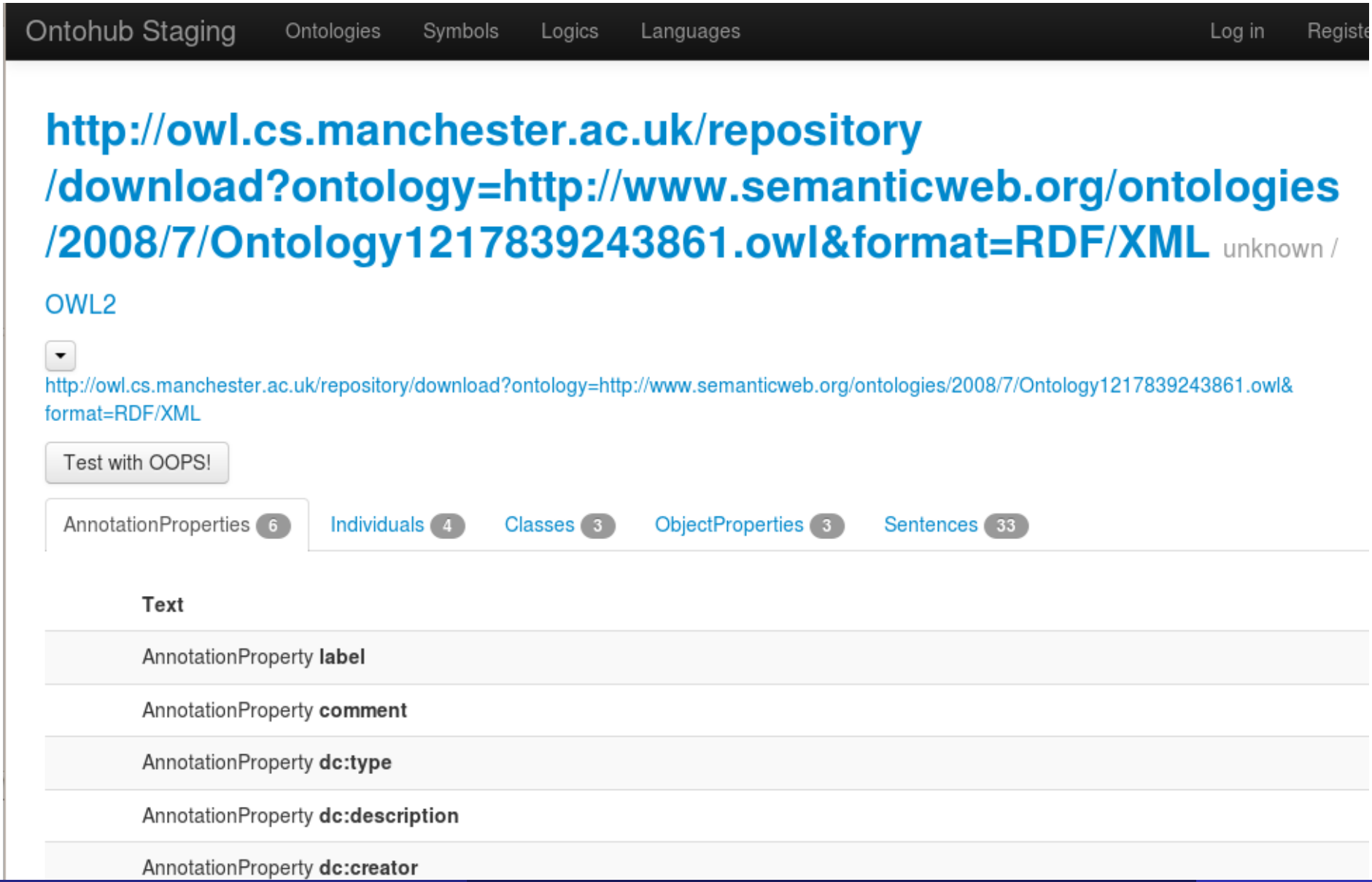
Involved Projects

- Open Ontology Repository (OOR), Ken Baclawski, Peter Yim: joint effort for ontology repository that significantly goes beyond BioPortal
  - Architecture and API available
  - current implementations: BioPortal, Ontohub
- Ontohub, Till Mossakowski, Danviel Couto Vale + 4 programmers: web-based repository engine for distributed heterogeneous (multi-language) ontologies
  - prototype available at ontohub.org
  - mid-term goal: follow the OOR architecture and API
- OOPS!, María Poveda Villalon: ontology evaluation tool for OWL, giving hints and listing pitfalls
  - XML-based web service available

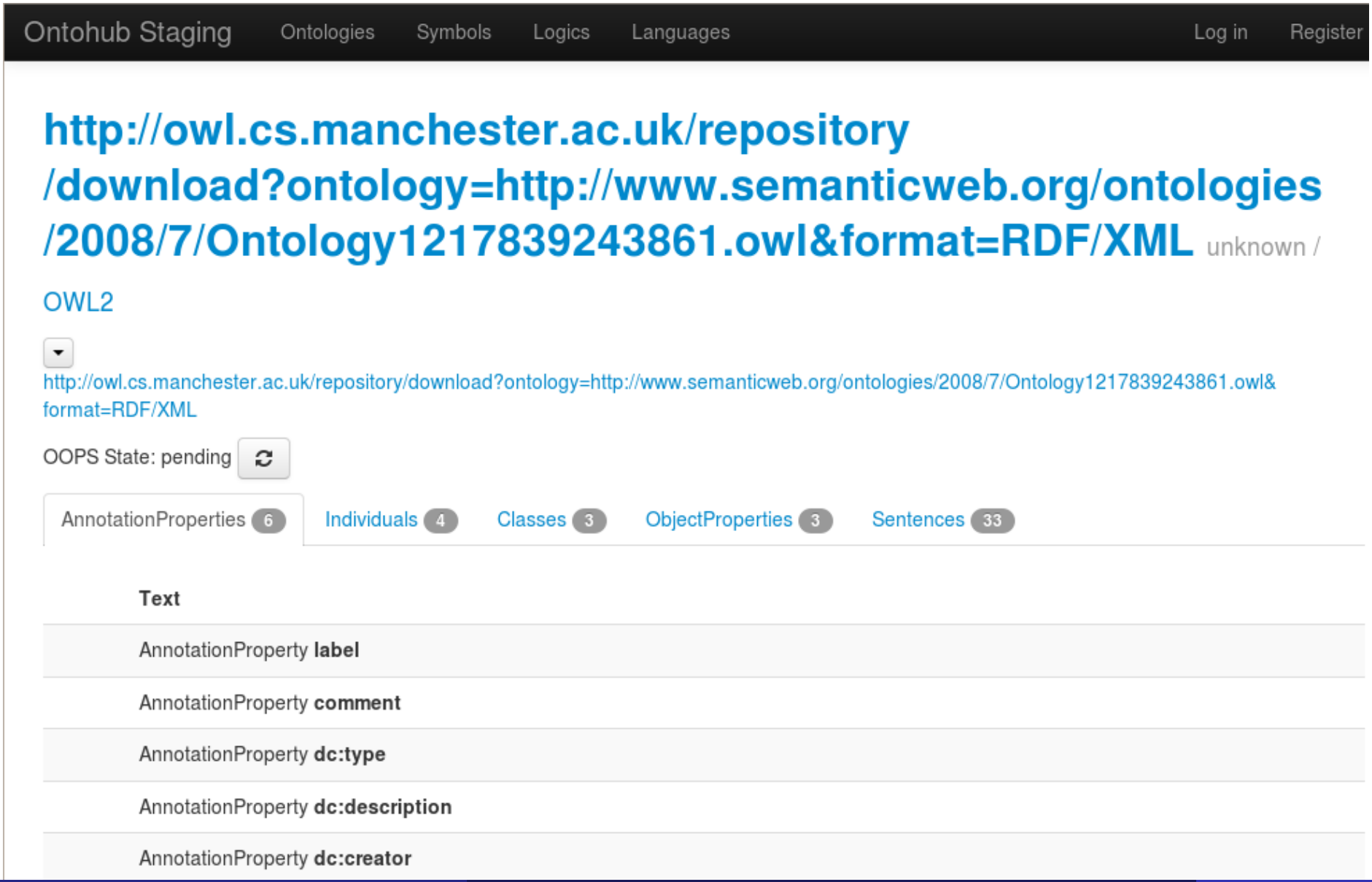
Results

1. integration of OOPS! output into the Ontohub.org display of classes and properties
   - code in the repository github.com/ontohub/ontohub providing this functionality
   - functionality provided online at staging.ontohub.org
2. architecture and webservices API integrating Ontohub, OOR, OOPS! and possible other software tools and environments
   - started with BioPortal API (= current OOR API)
   - integrated Ontohub's perspective: multiple ontology languages
   - integrated evaluation services (like OOPS!)

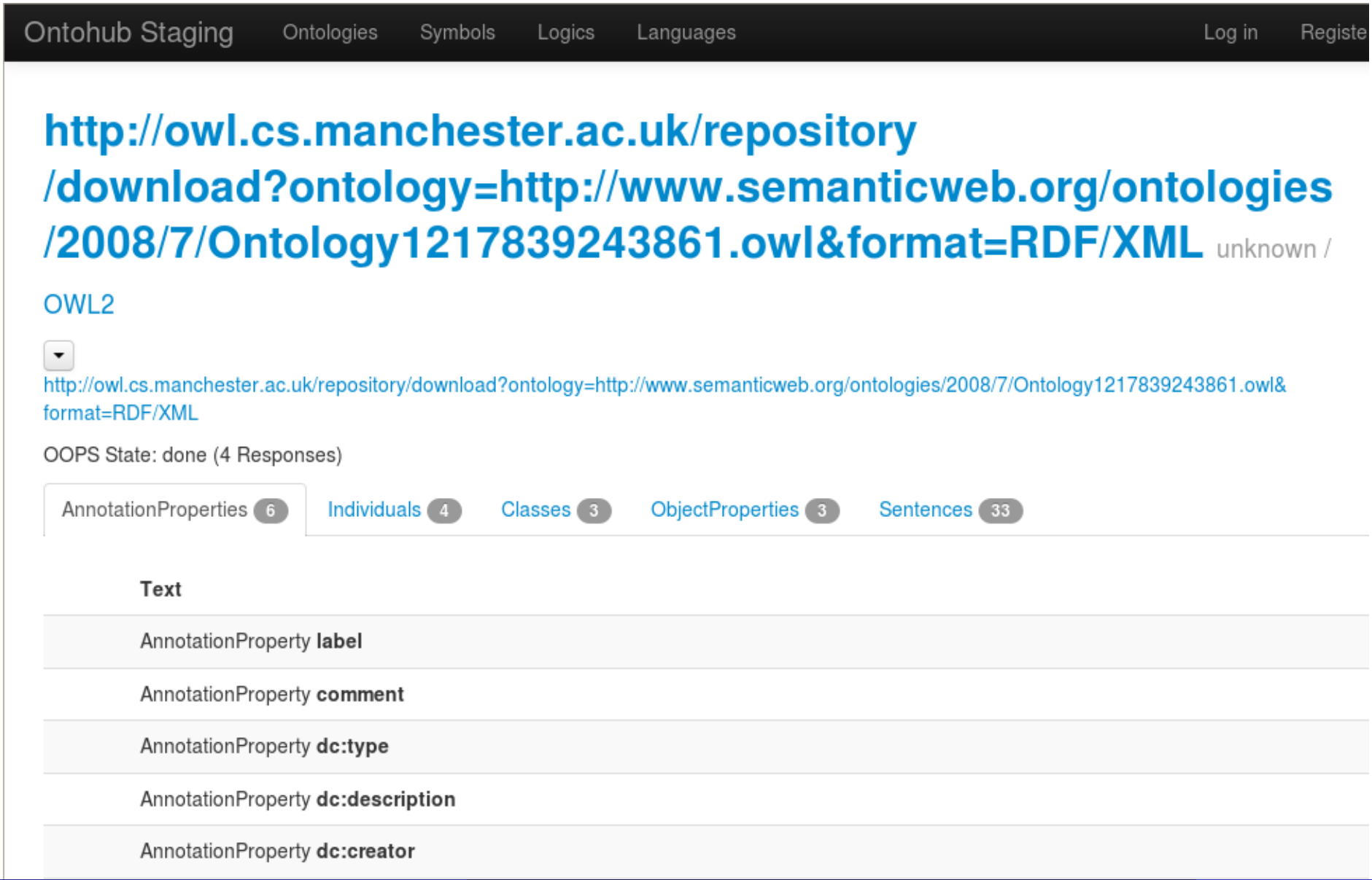
Demo of Ontohub-OOPS! integration



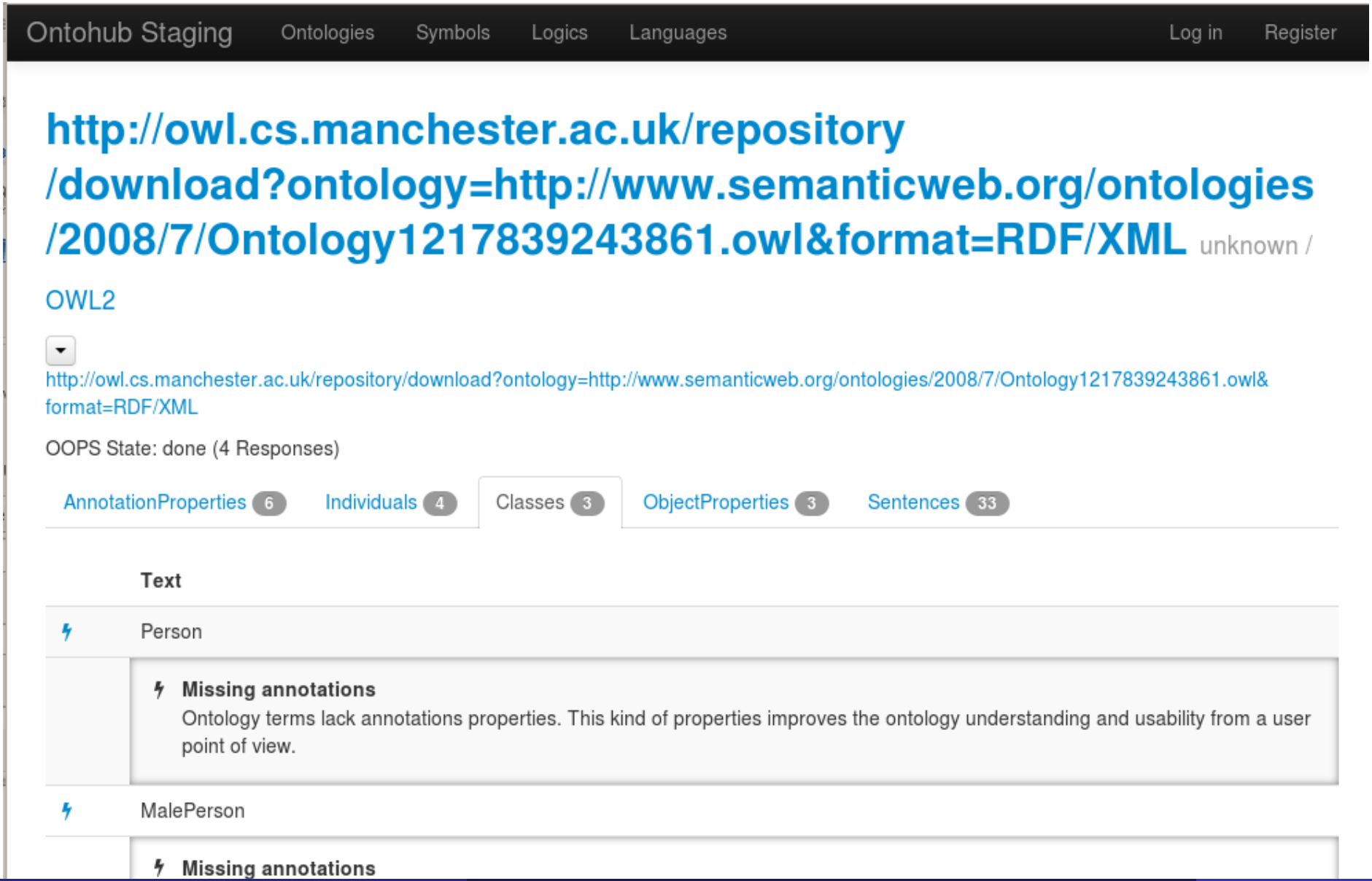
Demo of Ontohub-OOPS! integration



Demo of Ontohub-OOPS! integration



Demo of Ontohub-OOPS! integration



Ontohub's Notion of Ontology

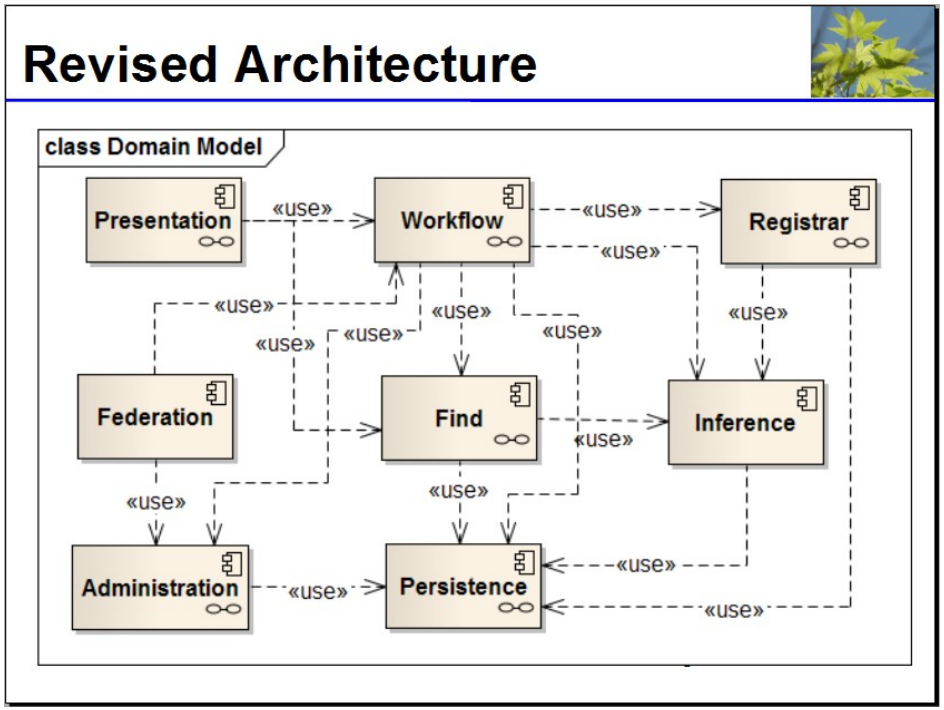
Ontohub's notion of an ontology is generic:

- a set of symbols
- each symbol has a kind:
  - in OWL: Class, ObjectProperty, DataProperty
  - in Common Logic: name, sequence marker
  - in first-order logic: predicate symbol, function symbol
- and a set of sentences (axioms, definitions, theorems)
- in some ontology language

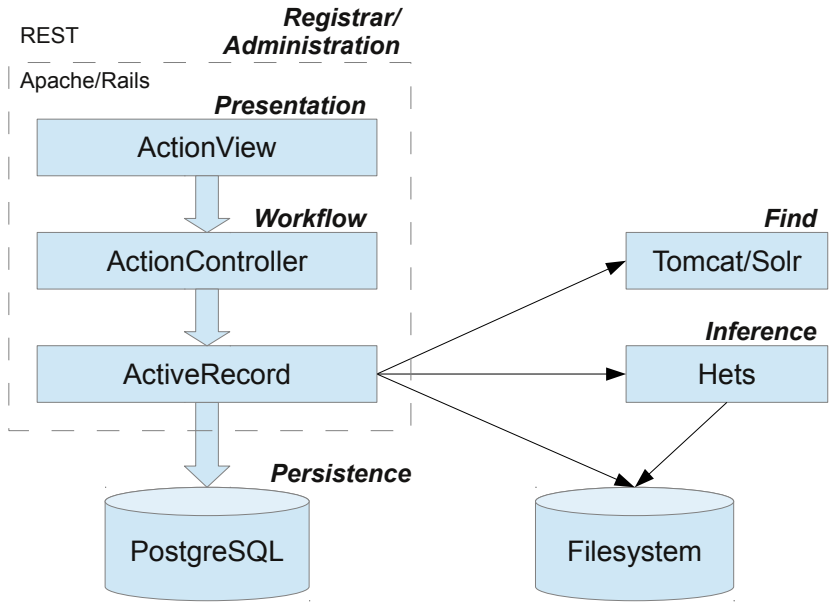
Semantics: theory of institutes (see OntoIOp/DOL)
All objects identified by IRIs, can have metadata and comments (only supported for ontologies so far).

Current Architecture

OOR:

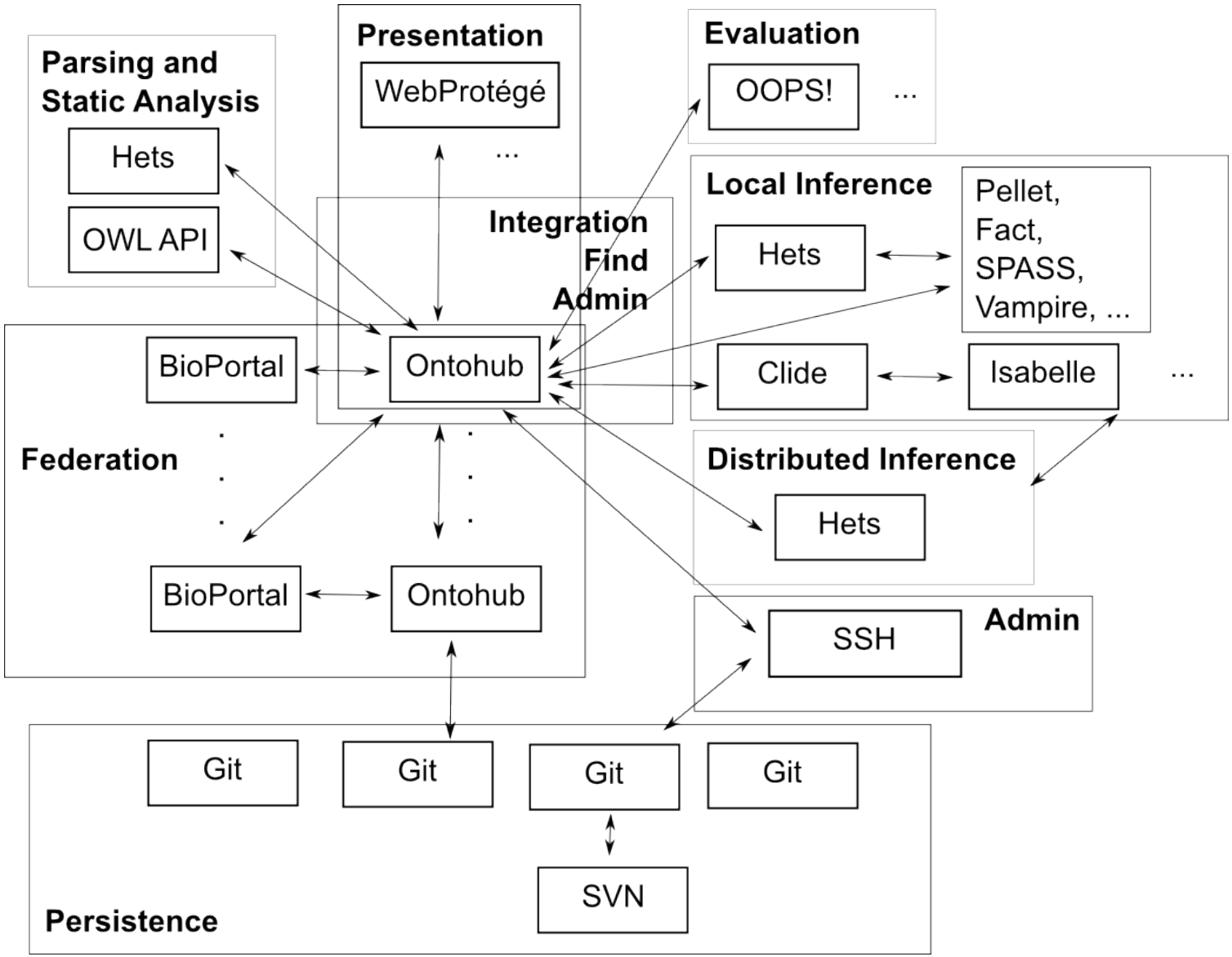


Ontohub:



Plan: adapt Ontohub successively to the OOR architecture, while keeping a running system

Proposed New Architecture



List of needed APIs

- federation: Ontohub↔Ontohub, Ontohub↔BioPortal,
  - take BioPortal API and generalise it to arbitrary logics
- automatic inference + evaluation: Ontohub↔Hets, Ontohub↔OOPS!
  - both Hets and OOPS! have APIs, abstract them
  - should be also used for direct communication with theorem provers (without Hets, via suitable wrappers)
- interactive inference: equip clide with API
  - `https://github.com/martinring/clide`
  - Martin Ring will join Ontohub team
- persistence and admin: bringit equips git with web interface
  - `https://github.com/eugenk/bringit`
  - Eugen Kuksa will join Ontohub team
- presentation: as first step, equip WebProtégé with API

Federation API: plan

take BioPortal API and generalise it to arbitrary logics. Examples:

- "List all the latest version of ontologies" $\Rightarrow$ keep
- "Get all terms using the specific ontology version id" $\Rightarrow$ generalize (term $\mapsto$ symbol)
- "List the latest version of all Views" $\Rightarrow$ drop for now (maybe generalize later on) (assumes class hierarchy)

Results are collected in a Google drive document:
http://tinyurl.com/onto-arch

Federation API: Logic Services

List all ontology languages	() ⇒ (list (language.id:name))
List all supported logics of language	(language.id) ⇒ (list (logic.id:name))
List all serializations of language	(langauge.id) ⇒ (list (serial.id:name))
List all logic translations	() ⇒ (list (logic-translation.id:name))
List all logic translations with source	(logic.id) ⇒ (list (logic-translation.id:name))
List all logic translations with target	(logic.id) ⇒ (list (logic-translation.id:name))
dto. for ontology language translations	

Federation API: Ontology Services

List all ontology latest versions	find Latest OntologyVersions
Get ontology latest version	index Ontology derived
Get ontology version metadata	index OntologyVersion
Get ontology symbols and sentences	find Ontology SymbolsAndSentences derived
Get ontology version symbols and sentences.	find OntologyVersion SymbolsAndSentences
Get ontology latest version metadata	get Ontology Metadata derived
Get an ontology version file	get OntologyVersion File
Get the ontology last version file	get Ontology File derived
Get metrics for an ontology version	get OntologyVersion Metrics
Get all ns prefixes of ontology	get Ontology Prefix
List all ontology categories	list Categories
List all ontology for a category	find Category Ontologies
List all ontology-using groups	list Groups

Federation API: Mapping Services

- Get a single mapping by its id. Return type of mapping and list of mapping elements
- Get a list of mappings filtered by parameters
- Get a list of mappings for a symbol
- Get a list of mappings between two symbols
- Get a list of mappings for an ontology
- Get a list of mappings between two ontologies
- Create a new mapping
- Update a Mapping
- Delete a Mapping

Federation API: Mapping Services (cont'd)

- Mapping Statistics
- Get Recent Mappings
- Get Number of Mappings To/From Given Ontology
- Get Number of Mappings to Terms in Given Ontology
- Get Number of Mappings by Users for a Given Ontology

Federation API: Parsing and Static analysis

- Get all kinds of symbols (for a given ontology language),
- Parse an ontology file and get all symbols and axioms (in a specific ontology language)
- Parse a DOL file and get all ontologies and links of the distributed ontology (this implicitly includes computation of ontologies specified by the DOL structuring constructs, e.g. ontology combinations)
- Translate an ontology along a logic or language translation

Local Inference API

- get available inference tools by name, language/logic, type (prover, model finder, conservativity checker, module extractor) and input parameters (including options)
- prove open goals in an ontology. Output: list of used axioms, proof, proof status using SZS ontology http://tinyurl.com/szsontology
- check consistency / find model of an ontology. Output: model, represented by symbols + axioms
- disprove open goals in an ontology. Output: see find model
- check conservativity of a link. Output: conservativity status (NotCons, DontKnow, Cons, Mono, Def)
- module extraction for an ontology w.r.t. a subsignature(=list of symbols) and an extraction algorithm

Distributed Inference API

- not clear yet
- idea: use Hets' development graph calculus
- central objects: distributed ontologies closed under the import/definition link relation
  - problem: this can become quite large
  - Should we therefore allow distributed ontologies to import other distributed ontologies in DOL?
- Distributed inference performs complex transformation on ontologies and links of a distributed ontology
  - Should therefore a new proof state of a distributed ontology constitute a new version, possibly with new version of the involved ontologies and links?
  - What happens with proof goals generated e.g. by interpretations? Can they spoil the target ontology? Better cerate a new ontology importing it?

API for Evaluation and other Services

we propose the following abstraction from the OOPS! API:

- input: ontology
  (well, OOPS! has more inputs, but these are not relevant here)
- output:
  - list of response elements of the following form:
  - type (for OOPS: pitfall, warning, suggestion)
  - code (an integer)
  - name
  - description
  - list of involved symbols

OOPS! outputs structured XML elements that may contain multiple n-ary relations between symbols (e.g. oops:MightBeEquivalentProperty and oops:MightBeEquivalentAttribute). We prefer to have only one such relation per response element.

Other Services (cont'd)

- Annotator Service
  - This service it specific to bio ontologies. How to generalise it to other domains? It seems that some (more static) list of service types and (more dynamically growing) list of actual services (conforming to these service types) would be useful. This of course also includes services like OOPS!
- Ontology Recommender
  - How to generalise this to ontologies in arbitrary languages?
- Resource Index Service
  - could be adapted, if "concept" is replaced by "symbol"
- Notes Service (Term Proposals and Comments)
- Logic-specific services
  - OWL specific services involving the class hierarchy These services could also be used for other languages if there is a suitable projection to OWL.