MANCHESTER 1824

The University of Manchester

The OWL API
A Java API for Working with OWL 2 Ontologies

Matthew Horridge

with special thanks to

Sean Bechhofer, Ron Alford, Nick Drummond, Birte Glimm, Olaf Noppens,
Ignazio Palmisano, Timothy Redmond, Thomas Schneider,
Evren Sirin, and Mike Smith

http://owlapi.sourceforge.net

OWL API

http://owlapi.sourceforge.net/index.html

Google

Home Download Source Support Documentation Reasoners Publications

The OWL API

The OWL API is a Java API and reference implmentation for creating, manipulating and serialising OWL Ontologies. The latest version of the API is focused towards OWL 2

The OWL API is open source and is available under the LGPL License

The OWL API includes the following components:

- An API for OWL 2 and an efficient in-memory reference implementation
- RDF/XML parser and writer
- OWL/XML parser and writer
- OWL Functional Syntax parser and writer
- Turtle parser and writer
- KRSS parser
- OBO Flat file format parser
- Reasoner interfaces for working with reasoners such as FaCT++, HermiT, Pellet and Racer

The original version of the API for OWL 1.0 was developed as part of the WonderWeb Project. Version 2.0.0 of the OWL API for was developed as part of the CO-ODE project and the TONES project. Version 3.0.0 is developed primarily at the University of Manchester.

The OWL API

2003 → 2007 → 2009

Version 1 | Version 2 | Version 3

OWL 1

High level syntax neutral interfaces based on the OWL Abstract Syntax

All changes applied through change objects

Interim OWL 2 (OWL 1.1)

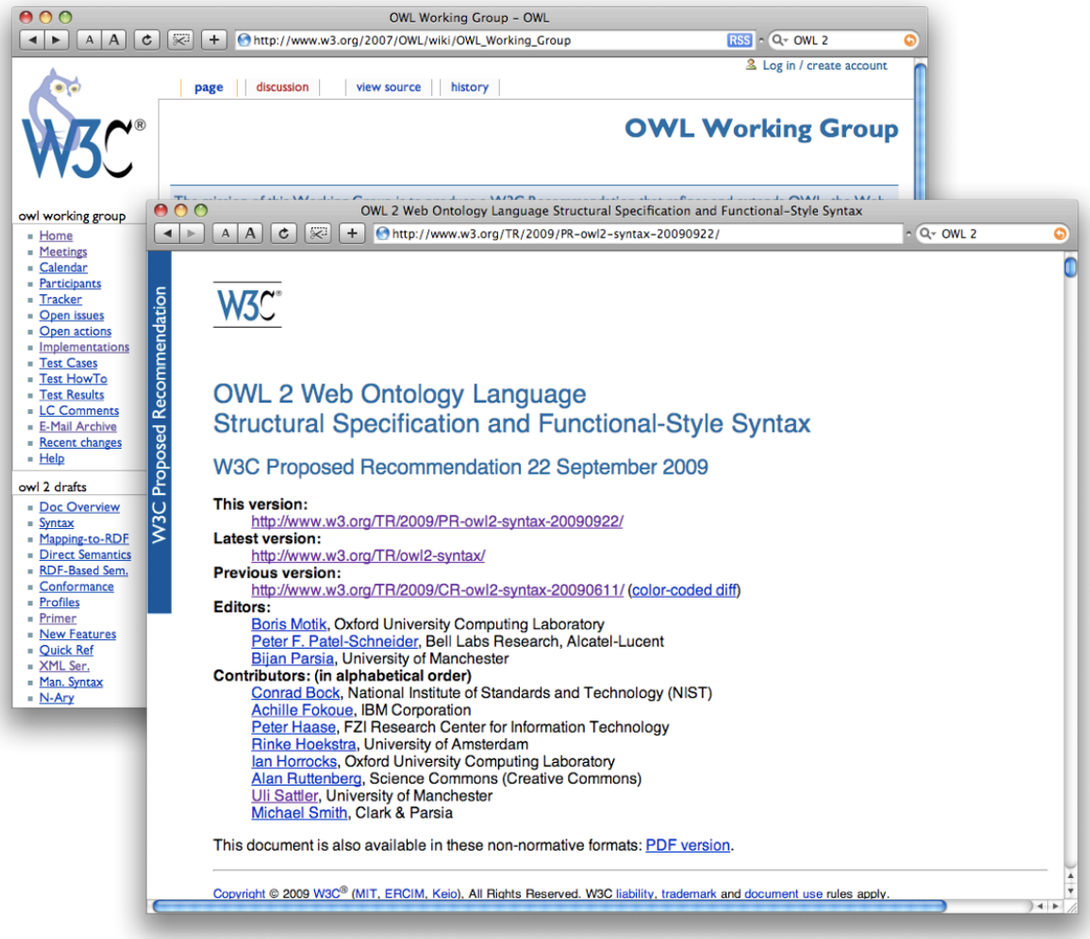
Shift from "frame" oriented model to axiom oriented model

OWL 2

Close alignment with OWL 2 specification

Addition of more convenience functionality

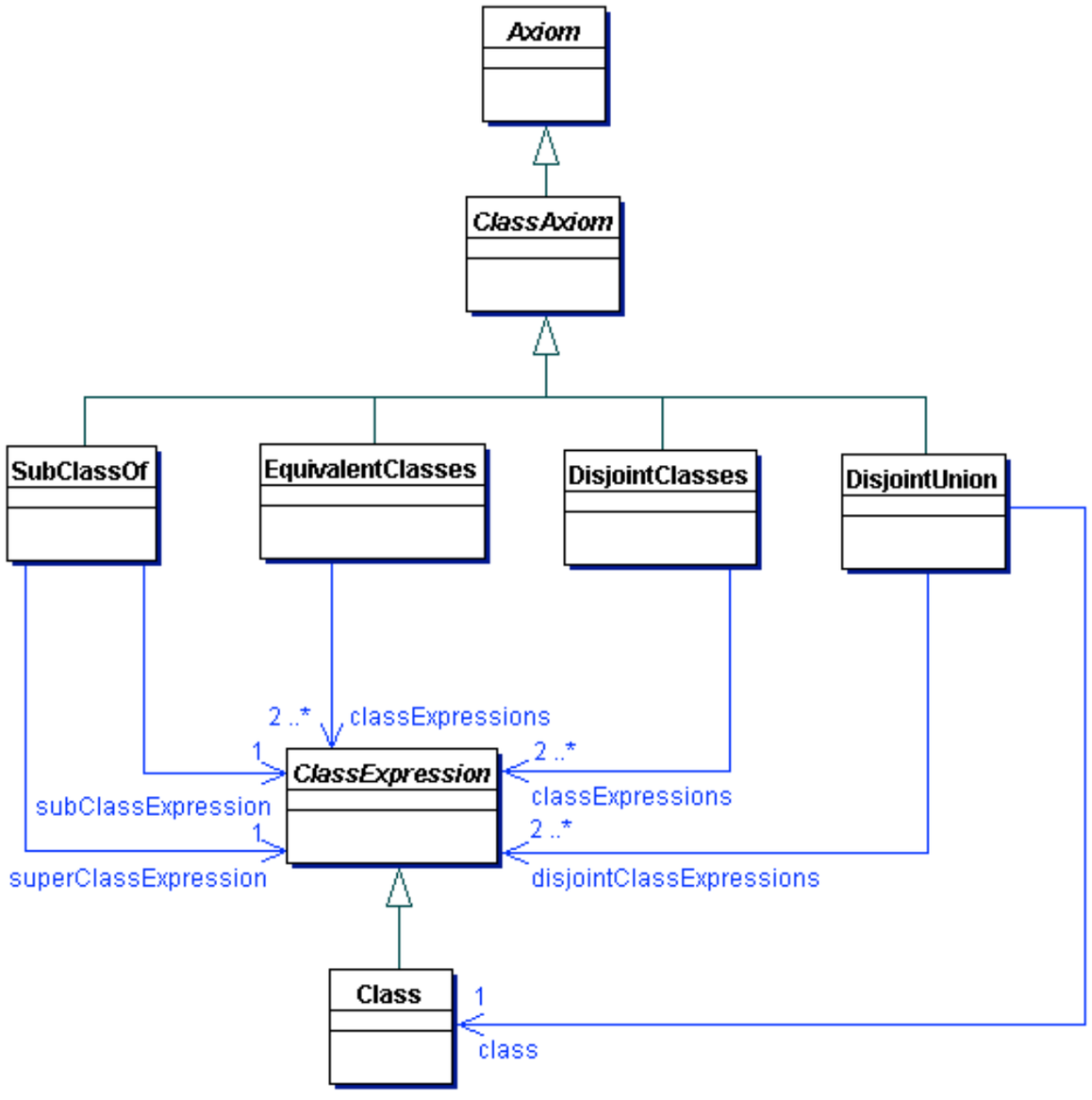
Design Philosophy

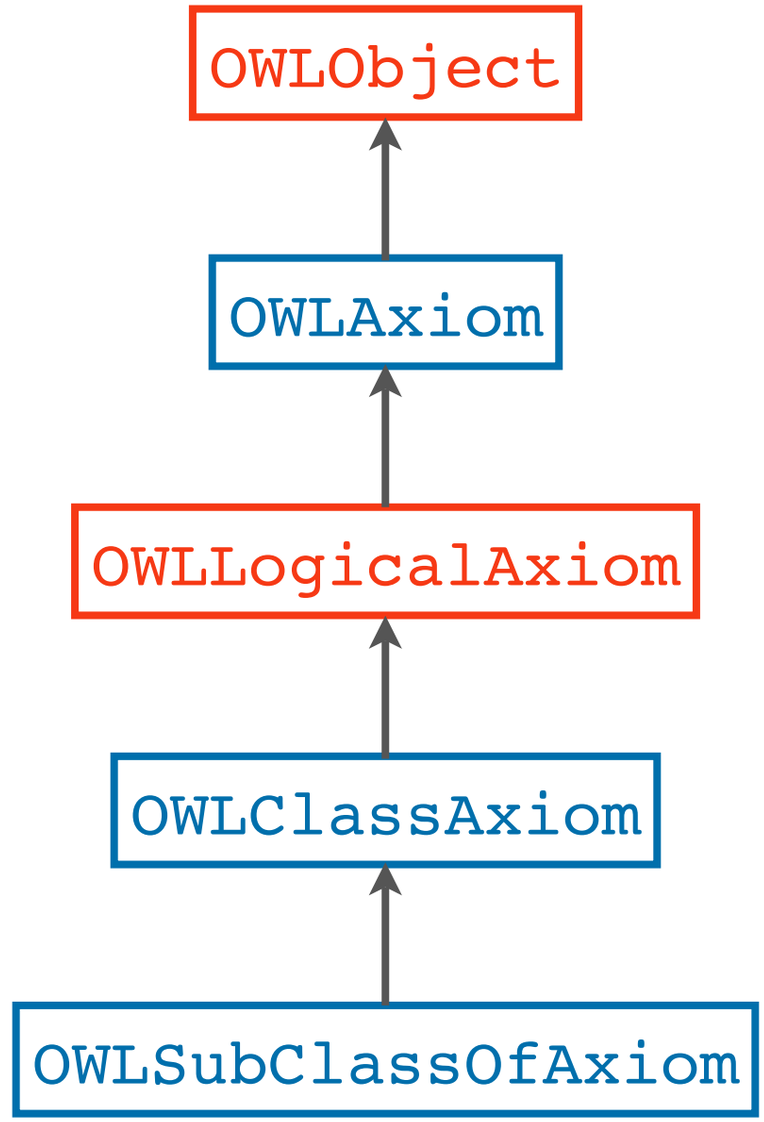


OWL API

Java interfaces
for manipulating
OWL 2 ontologies

From the Structural Specification to Java Interfaces





Other Machinery

Ontology management

Creation, loading and saving of ontologies

Change support

Change objects for axiom addition and removal, listener support

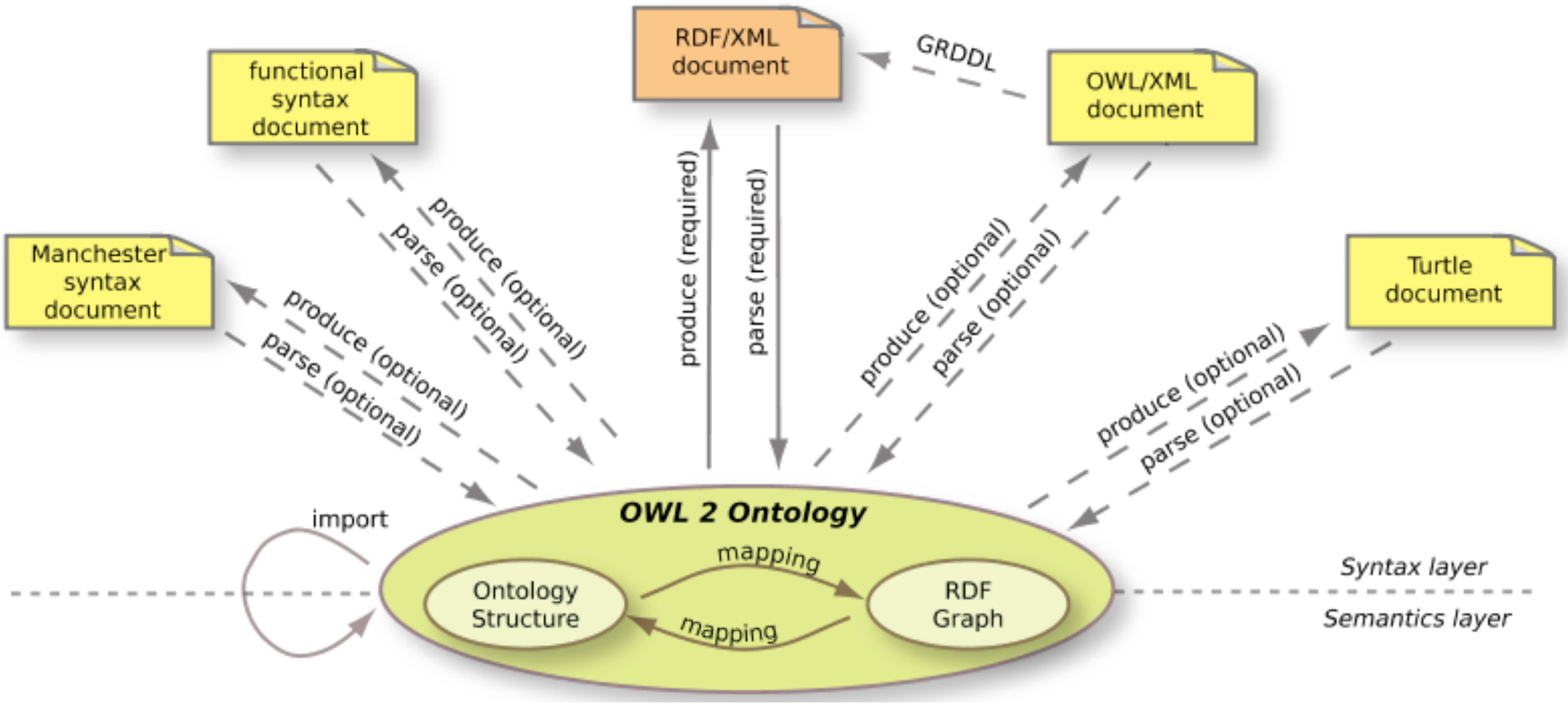
Common task support

Syntactic validation, metrics, normalisation

Reasoning support

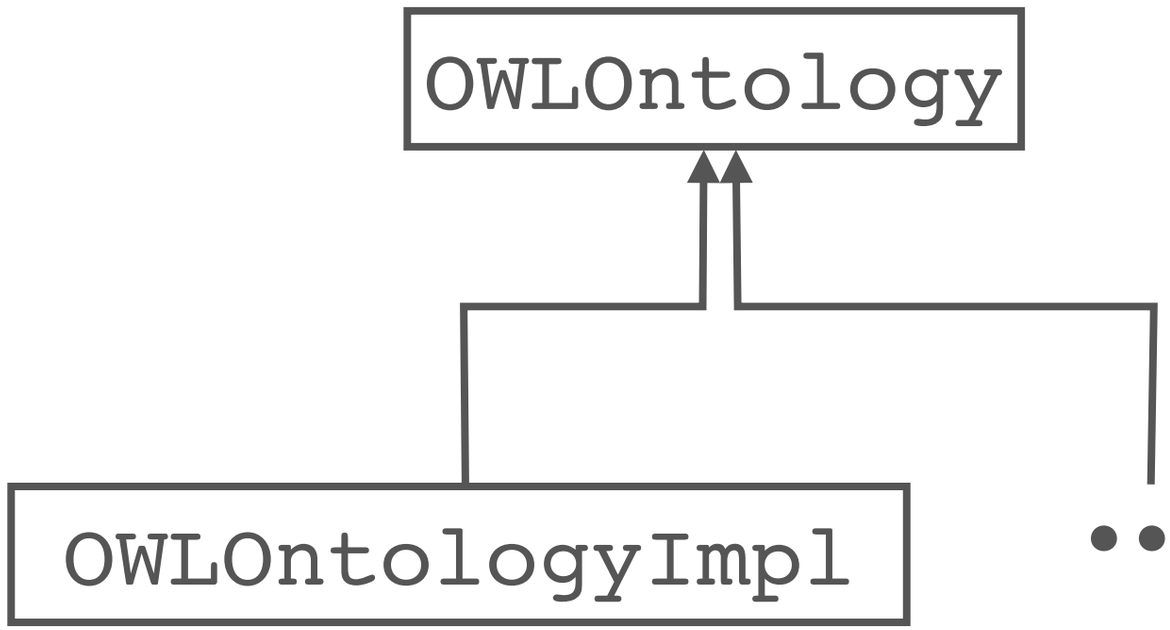
Common reasoner interfaces and reasoner operations

Concrete Syntaxes



(Courtesy of OWL 2 Web Ontology Language Document Overview)

Alternative Storage



Main memory reference implementation

`http://owldb.sourceforge.net/`

Reasoning

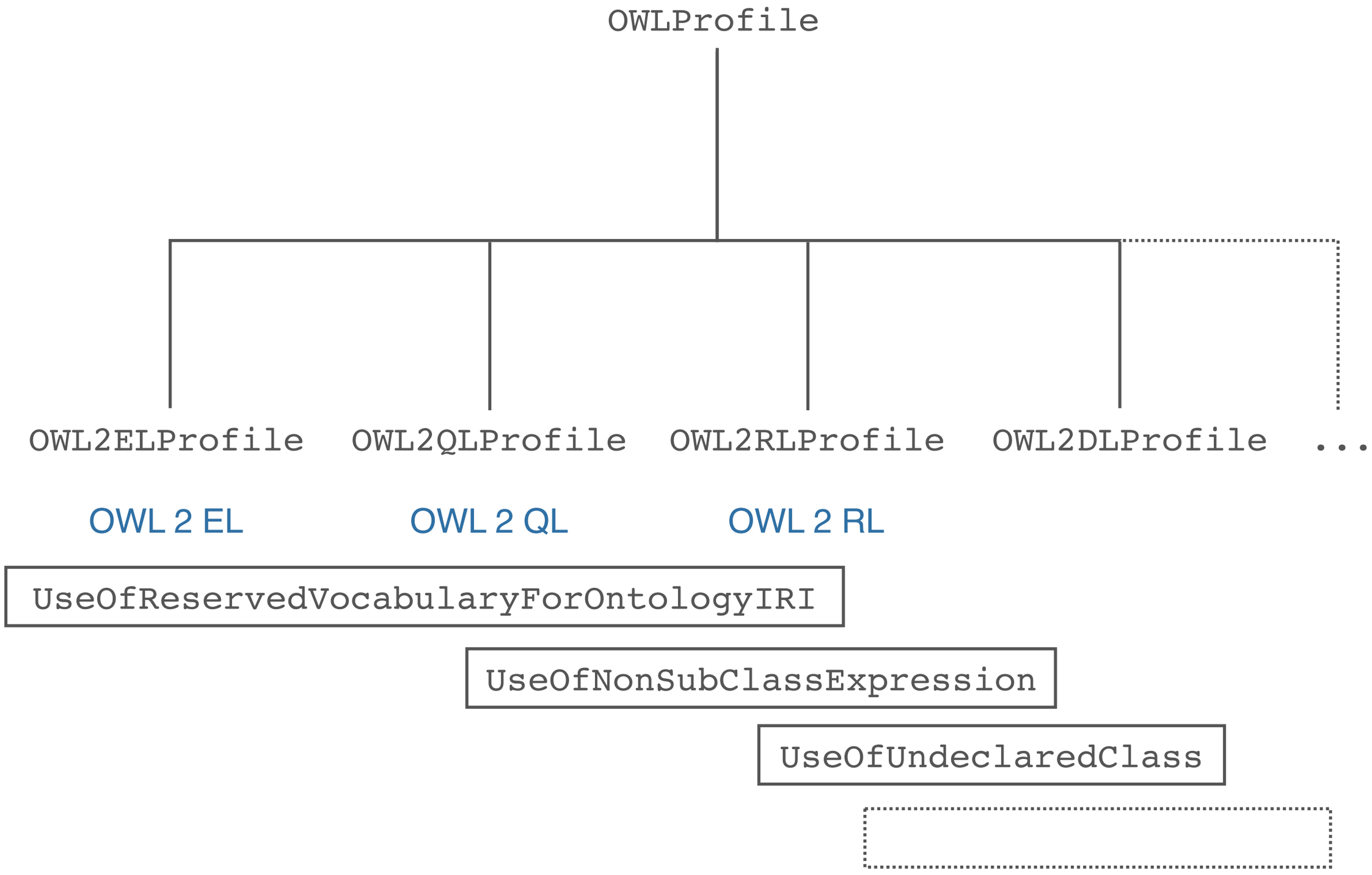
```
public interface OWLReasoner {

    /**
     * Asks the reasoner to interrupt what it is currently doing.  An InterruptedException will be thrown in the
     * thread that invoked the last reasoner operation.  The OWL API is not thread safe in general, but it is likely
     * that this method will be called from another thread than the event dispatch thread or the thread in which
     * reasoning takes place.
     */
    void interrupt();

    /**
     * A convenience method that determines if the set of reasoner axioms (the set of axioms returned by
     * the {@link #getAxioms()} method) is consistent.
     * @return <code>true</code> if the set of axioms is consistent,
     * or <code>false</code> if the set of axioms is inconsistent.
     * @throws InterruptedException if the reasoning process was interrupted for any particular reason (for example if
     * reasoning was cancelled by a client process)
     */
    boolean isConsistent() throws InterruptedException;

    /**
     * A convenience method that determines if the specified class expression is satisfiable with respect to the
     * set of reasoner axioms (the set of axioms returned by the {@link #getAxioms()} method)
     * @param classExpression The class expression
     * @return <code>true</code> if classExpression is satisfiable with respect to the set of axioms, or
     * <code>false</code> if classExpression is unsatisfiable with respect to the axioms.
     * @throws InconsistentOntologiesException if the reasoner's axiom set is inconsistent
     * @throws EntitiesNotInSignatureException if the signature of the classExpression is not contained within the signature
     * of the reasoner's axiom set.
     * @throws ExpressivenessOutOfScopeException If the class expression contains constructs that are out of the scope
     * of expressiveness that is supported by this reasoner.
     * @throws InterruptedException if the reasoning process was interrupted for any particular reason (for example if
     * reasoning was cancelled by a client process)
     */
    boolean isSatisfiable(OWLClassExpression classExpression) throws InterruptedException;
```

Profiles API



MANCHESTER 1824

The University of Manchester

OWL 2 Validation Report

Summary

Ontology and imports closure is NOT in OWL 2 EL profile

Imports Closure

Ontology IRI	Physical URI
http://www.co-ode.org/ontologies/pizza/pizza.owl	http://www.co-ode.org/ontologies/pizza/pizza.owl

Detailed report

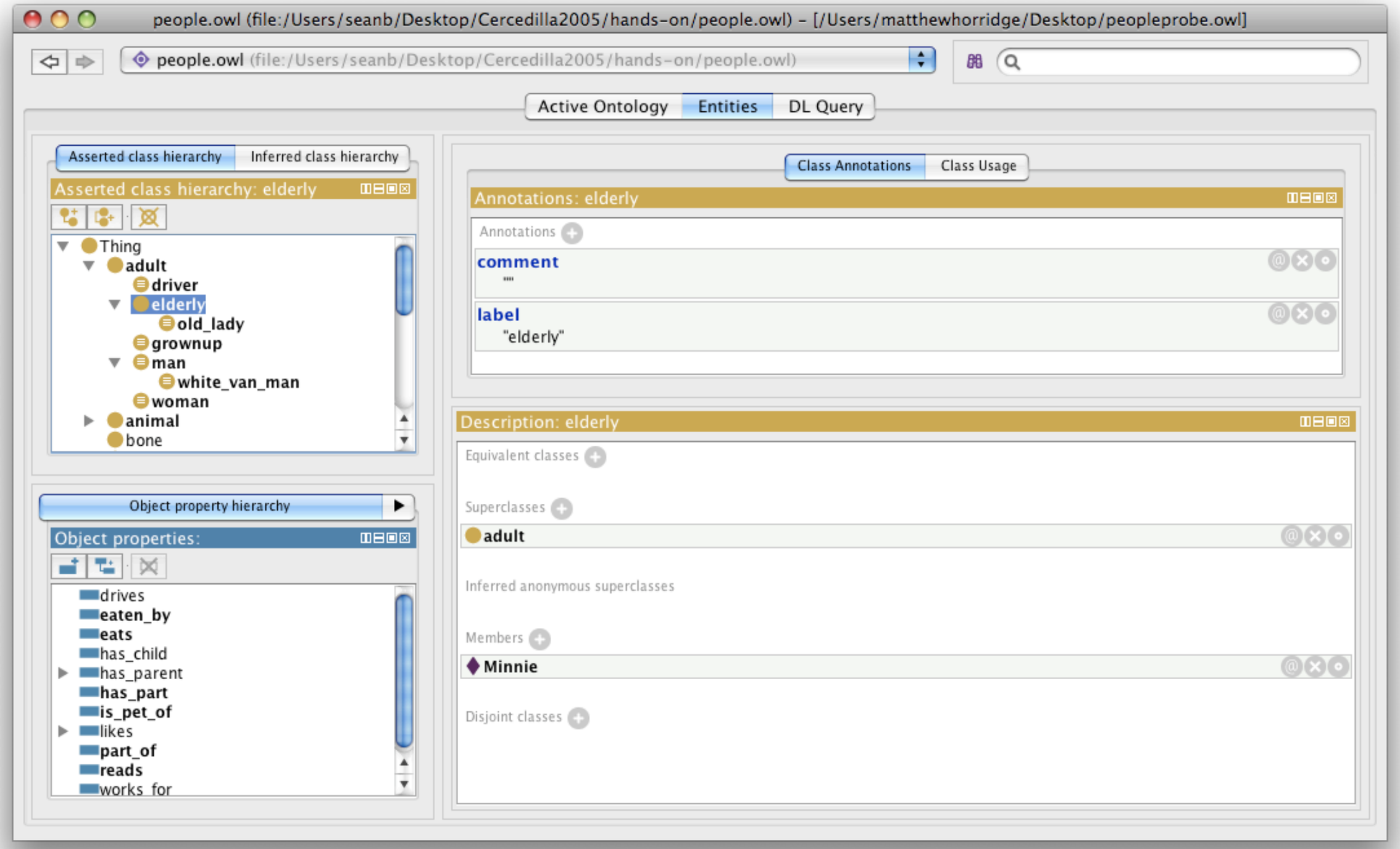
Use of unsupported class expression

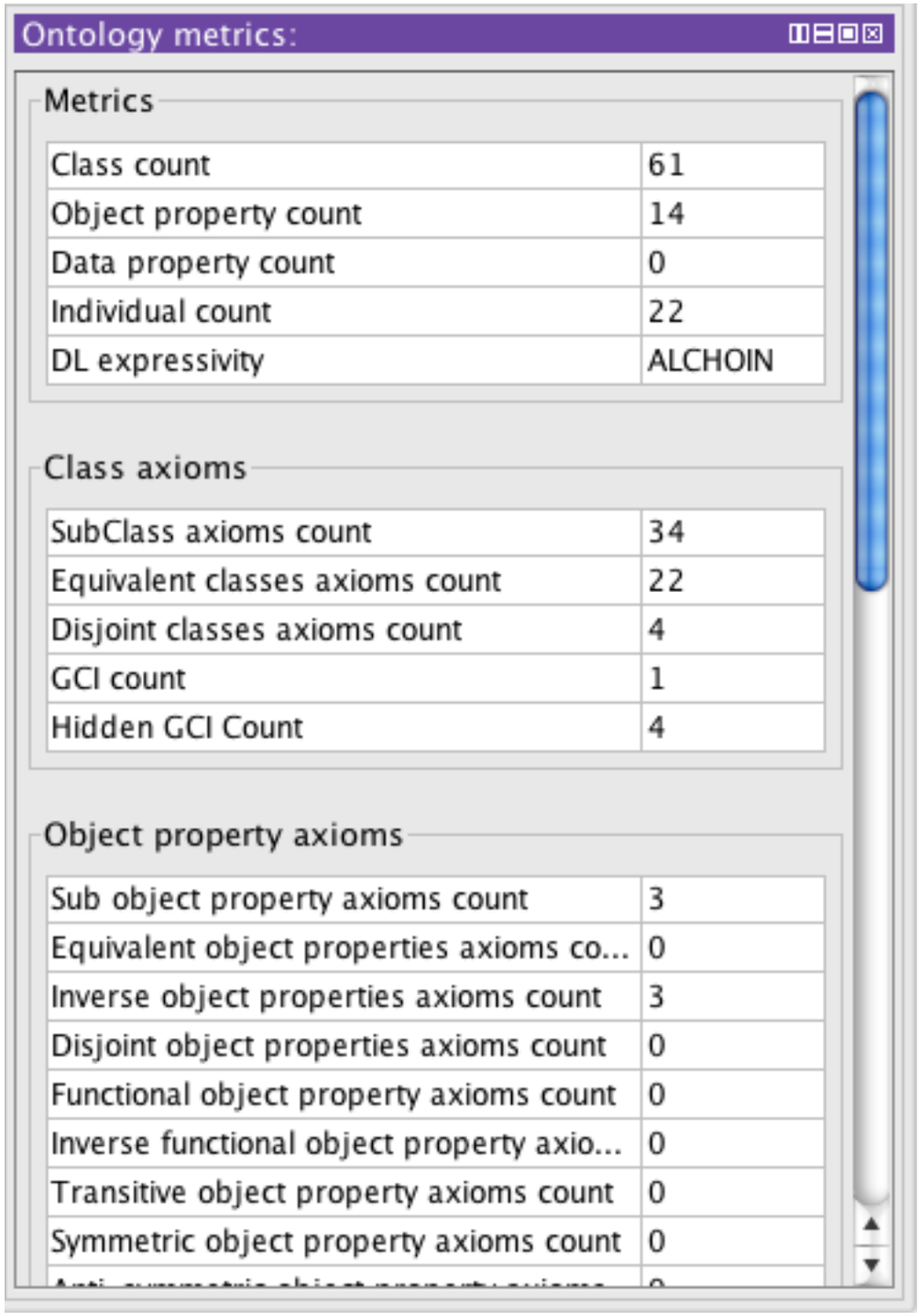
SubClassOf(:Rosa ObjectAllValuesFrom(:hasTopping ObjectUnionOf(:GorgonzolaTopping :MozzarellaTopping :TomatoTopping)))

Use of unsupported axiom

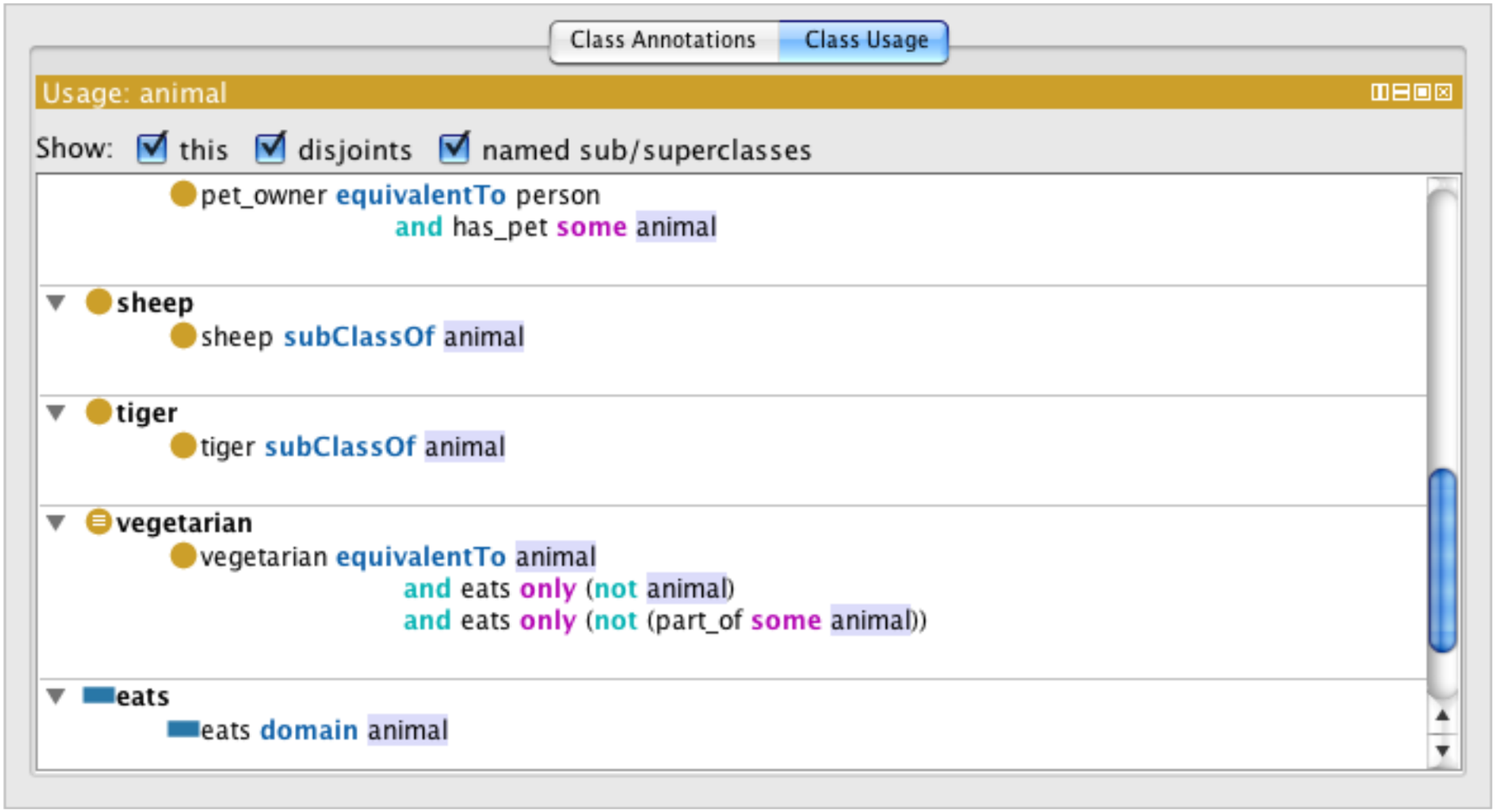
FunctionalObjectProperty(:hasSpiciness)

Examples of Use

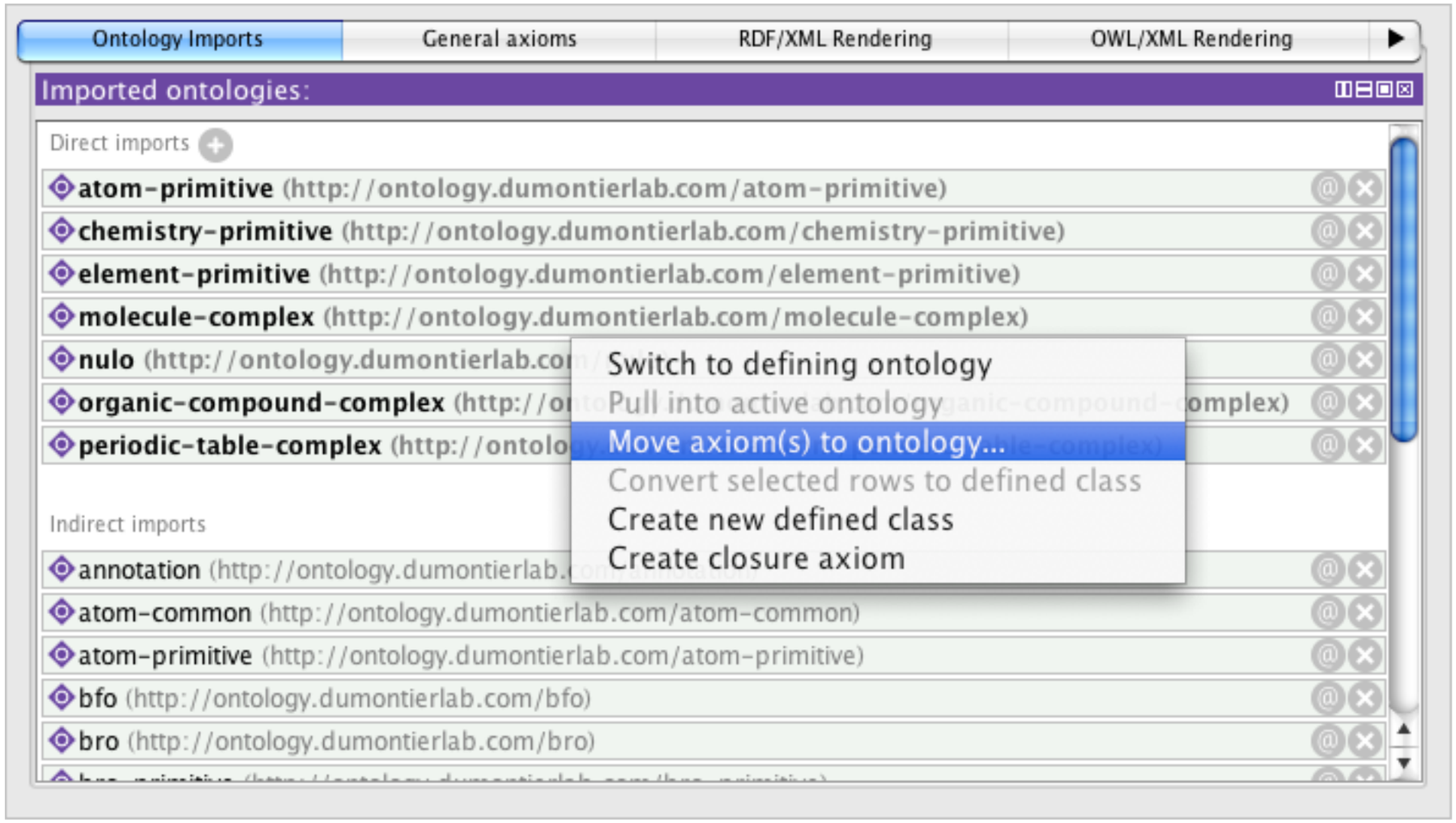




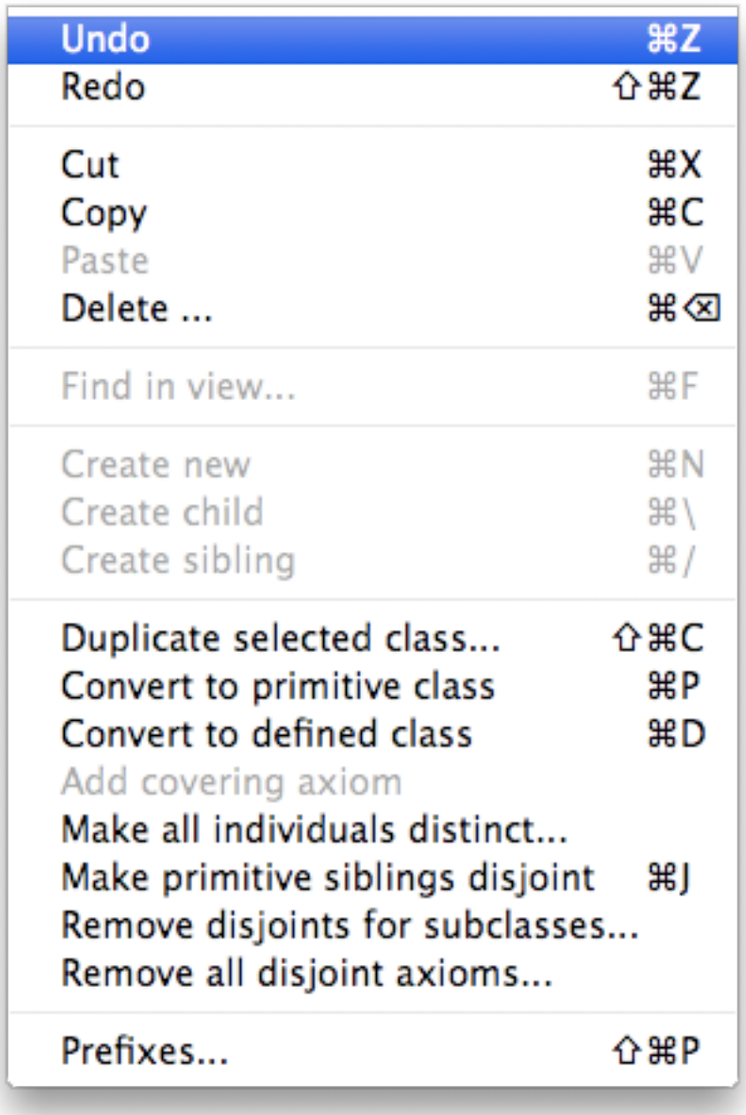
Metrics API for ontology metrics view



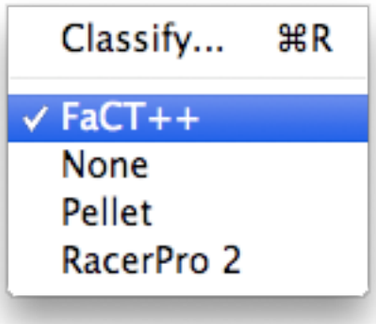
Axiom retrieval by signature and type for usage views



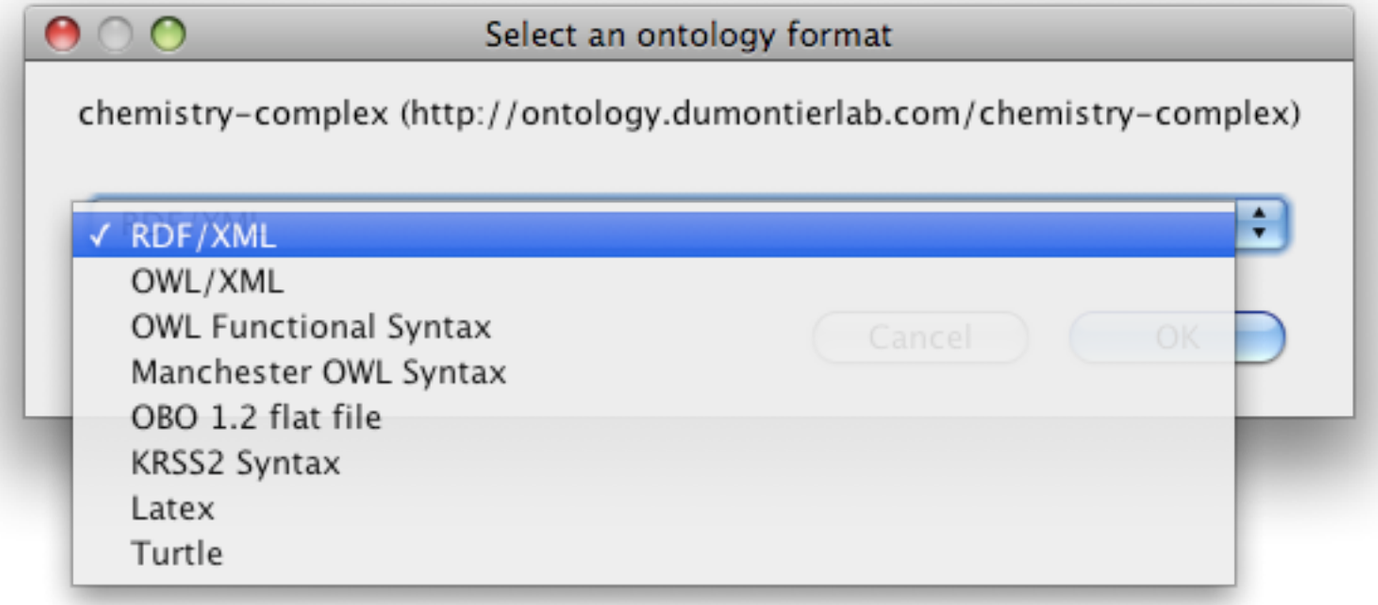
Working with multiple ontologies in an imports closure



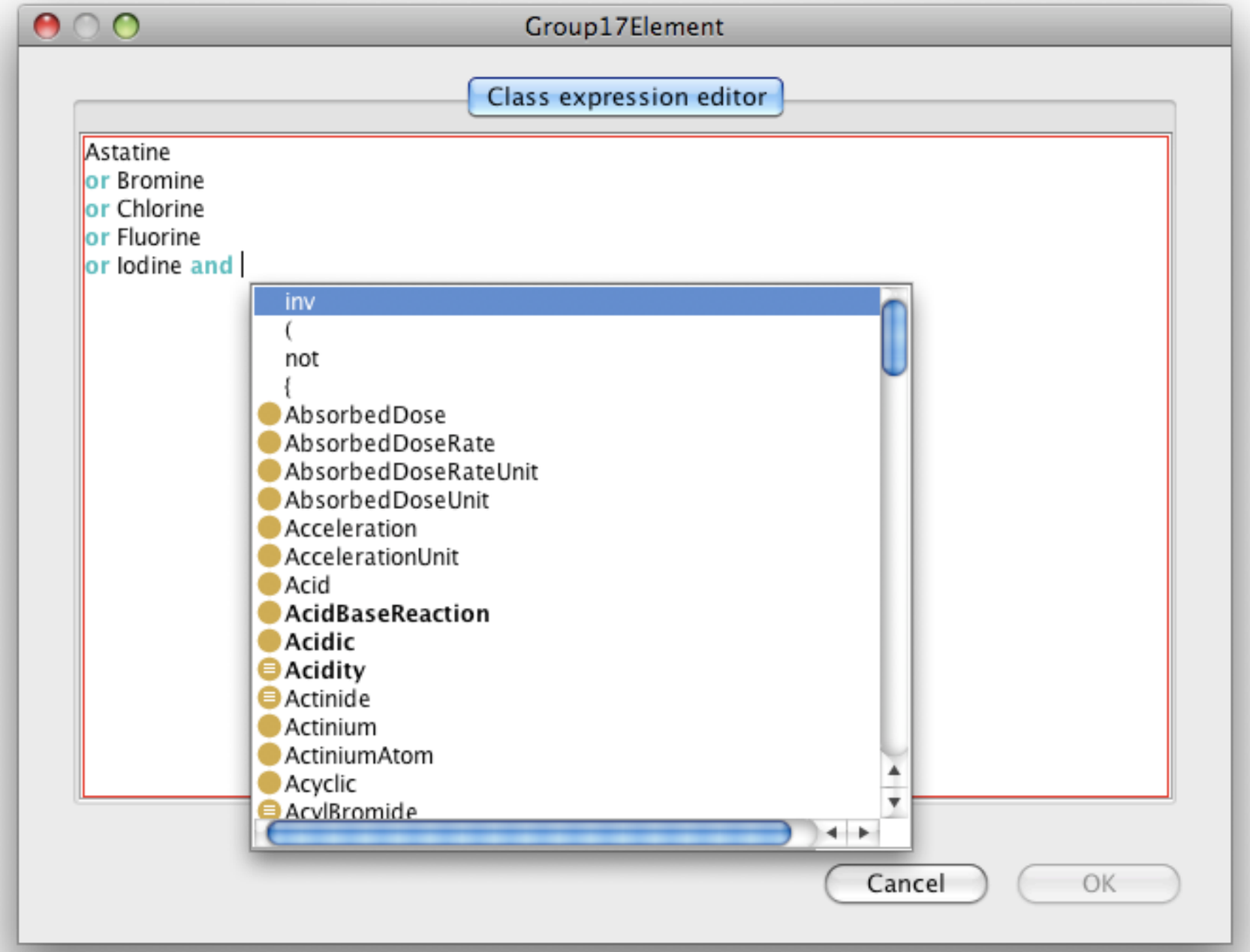
Change objects used for undo/redo



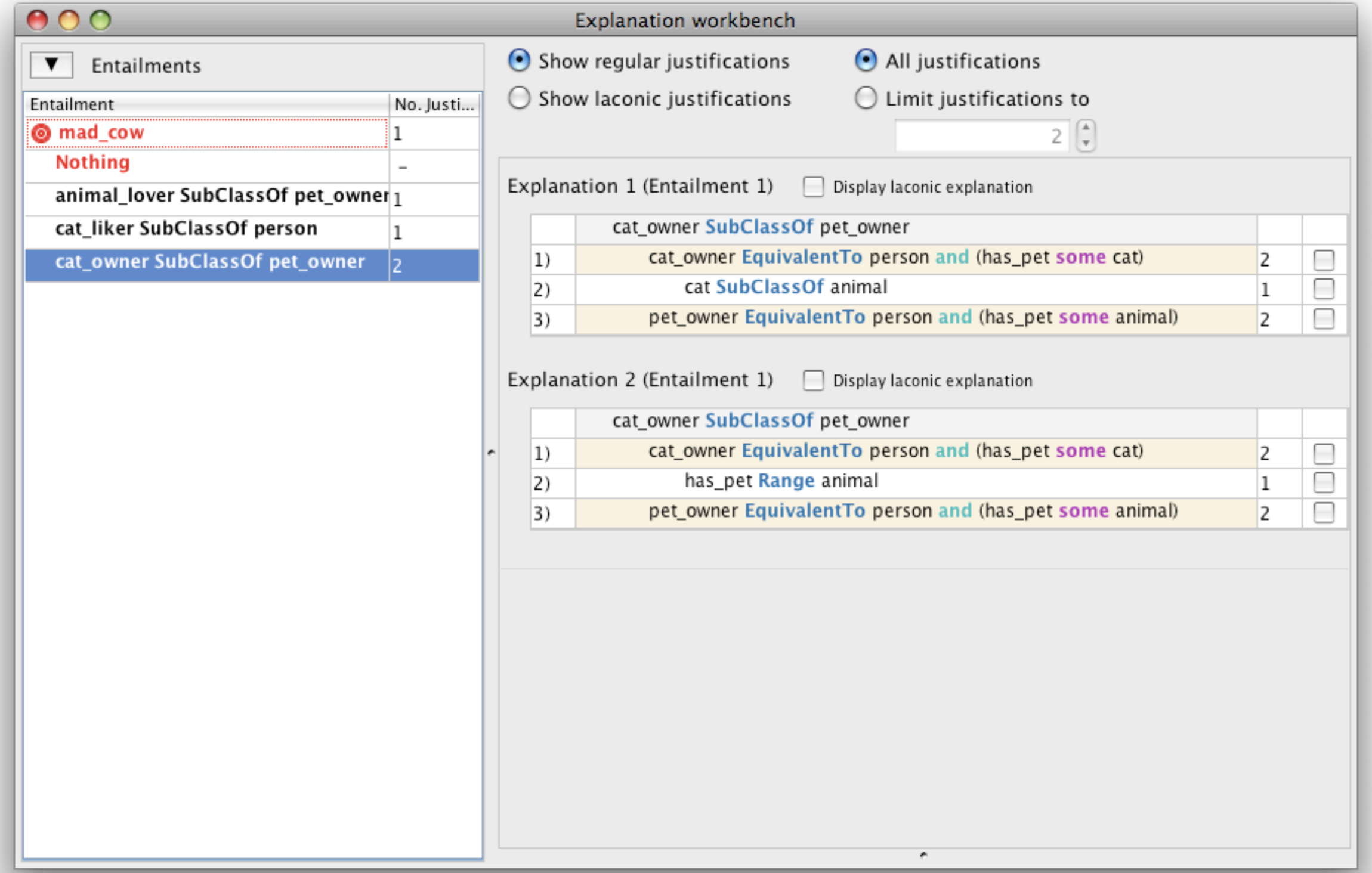
Common reasoner interface



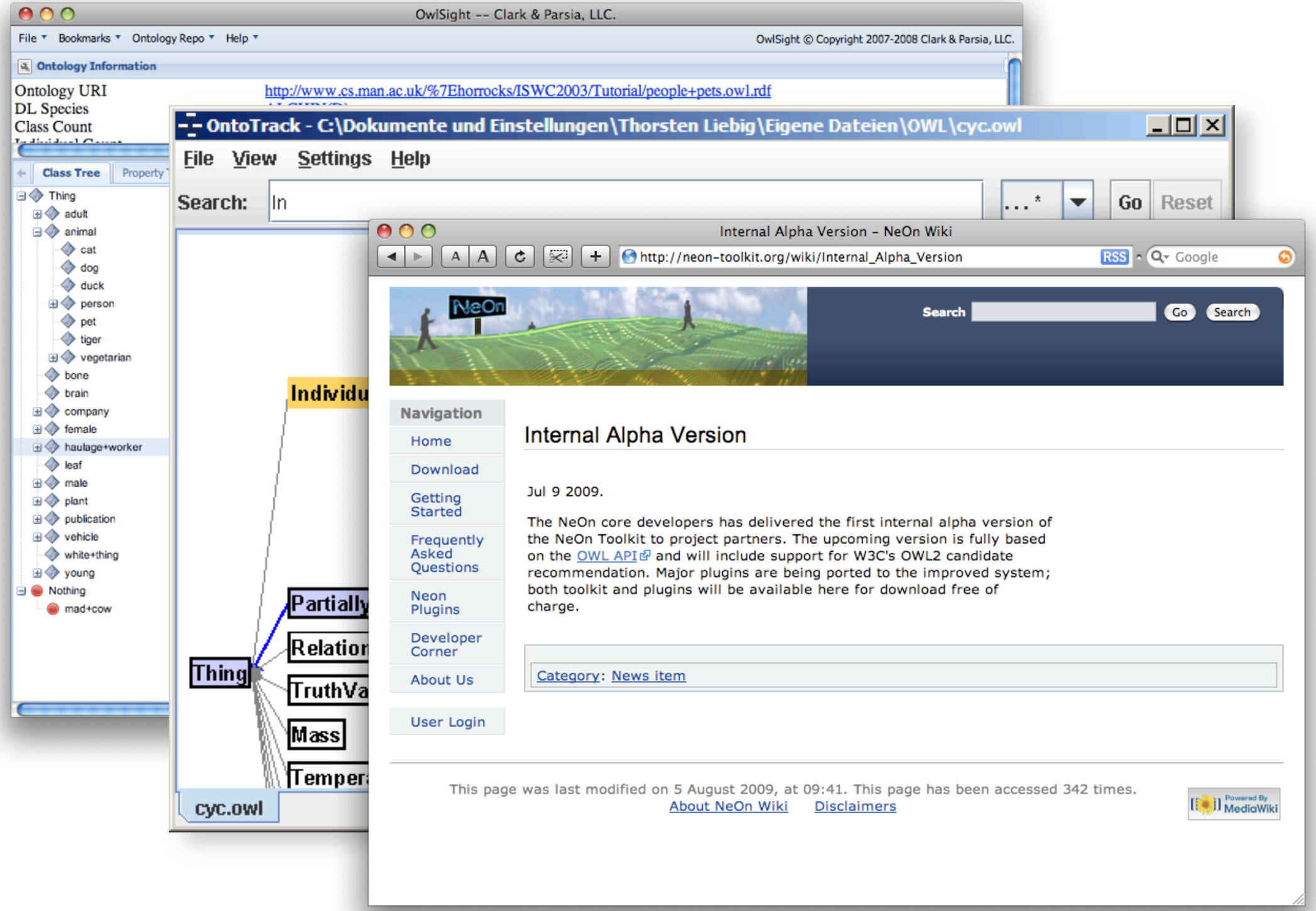
Out of the box serialisation support for multiple file formats



Use of Manchester Syntax parser to provide editor autocompletion and syntax checking



Axiom oriented model simplifies many tasks



Widely used in many other tools

Acknowledgements

Sean Bechhofer

Acknowledgements

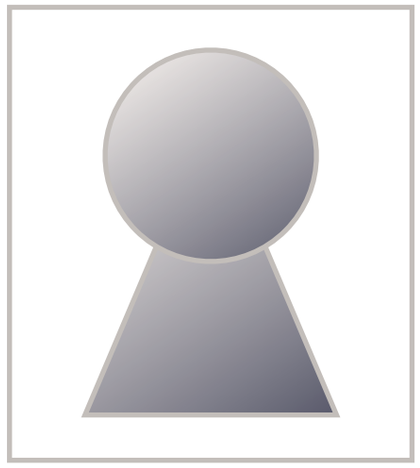
Birte Glimm

Boris Motik

Tim Redmond

Ignazio Palmisano

Acknowledgements



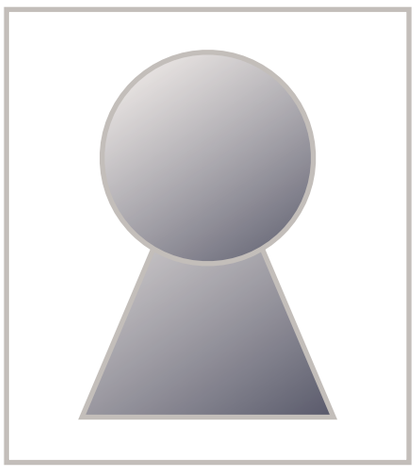
Ron Alford

Nick Drummond

Olaf Noppens

Thomas Schneider

Evren Sirin



Mike Smith