

# Ampersand and the Semantic Web

The Ampersand Conference 2015

Lloyd Rutledge

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



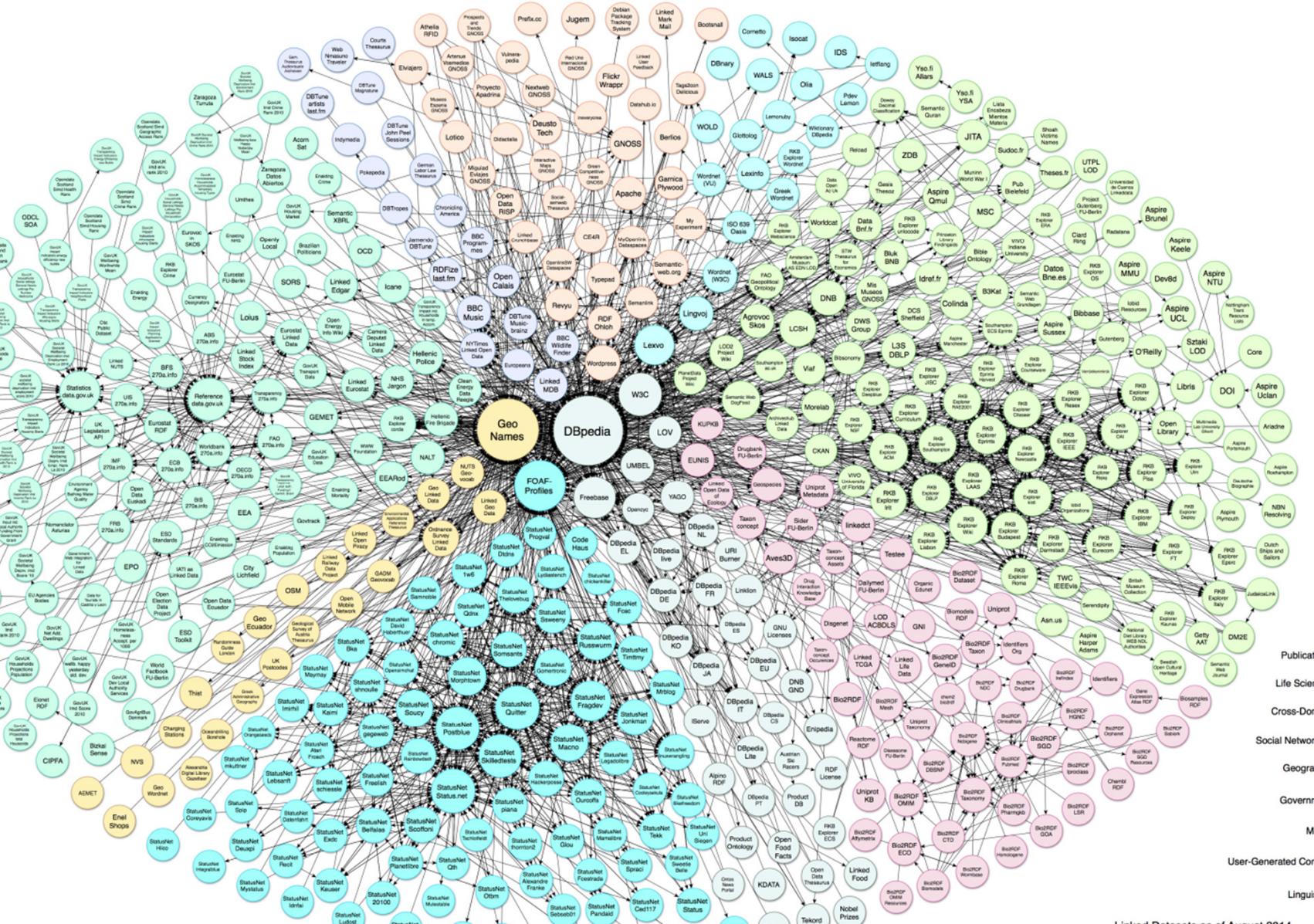
# The Semantic Web

- Billions and billions of data units
- Triples (subject-predicate-object) of URI's
- Your data readily integrated with that of others
- Common data models and vocabularies
- The whole world can share it as one common pool of data



Open Universiteit  
[www.ou.nl](http://www.ou.nl)





# ***Given a scenario you'd implement with Ampersand, how would you implement it differently on the Semantic Web?***

- Why
  - Prototype for Semantic Web system
  - Reuse of Semantic Web tools
  - Comparative analysis of layer of logic
  - Use of common data and models ... and rules?
- Where: research, education, industry



# Ampersand vs Semantic Web: *Logic*

- **Ampersand**
  - Relation algebra!
- **Semantic Web**
  - *RDFS* : Description Logic (data modeling) for inferencing
  - *OWL*: Enables inconsistency
  - *SWRL*: First order / Horn clauses for own rules
  - *SPARQL*
    - SQL-like query language
    - Only place with *negation*, but then as external built-ins
  - *SWRL-like ontology for relation algebra extensions to the Semantic Web???*

Open Universiteit  
[www.ou.nl](http://www.ou.nl)

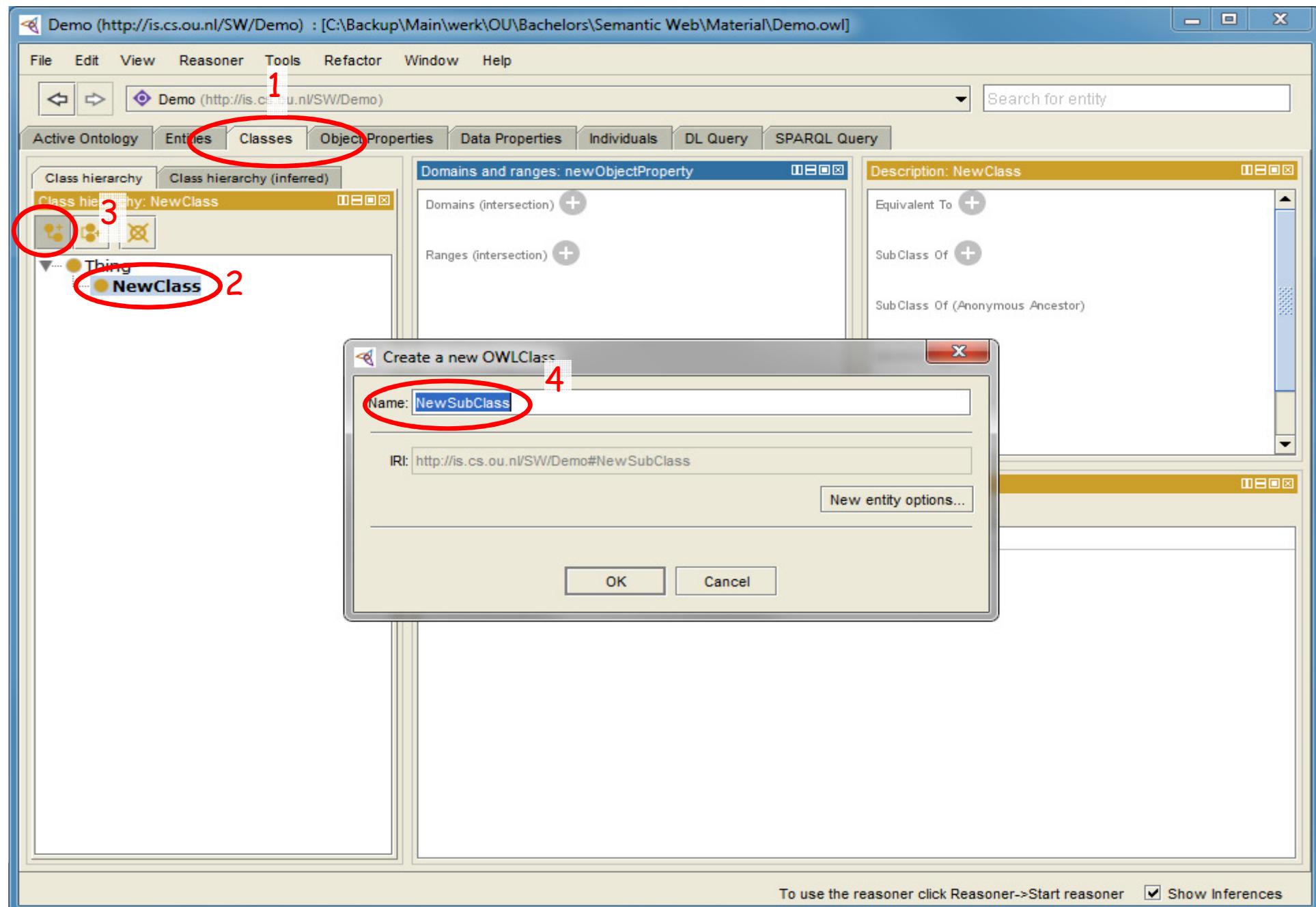


# The Semantic Web can't say “No”

- No negation as failure
  - Can't say something *isn't* true (or ask or conclude based on it)
- No negated existential quantification
  - Can't say something *doesn't* exist
- Open World Assumption
  - Current data collection may be *incomplete*
- No unique naming assumption
  - Different URI's can mean the *same thing*
- Why not?
  - Helps handle *incomplete* data
  - Scalability, robustness, etc.



# Concept = class



To use the reasoner click Reasoner->Start reasoner  Show Inferences

# Generalization = subclass

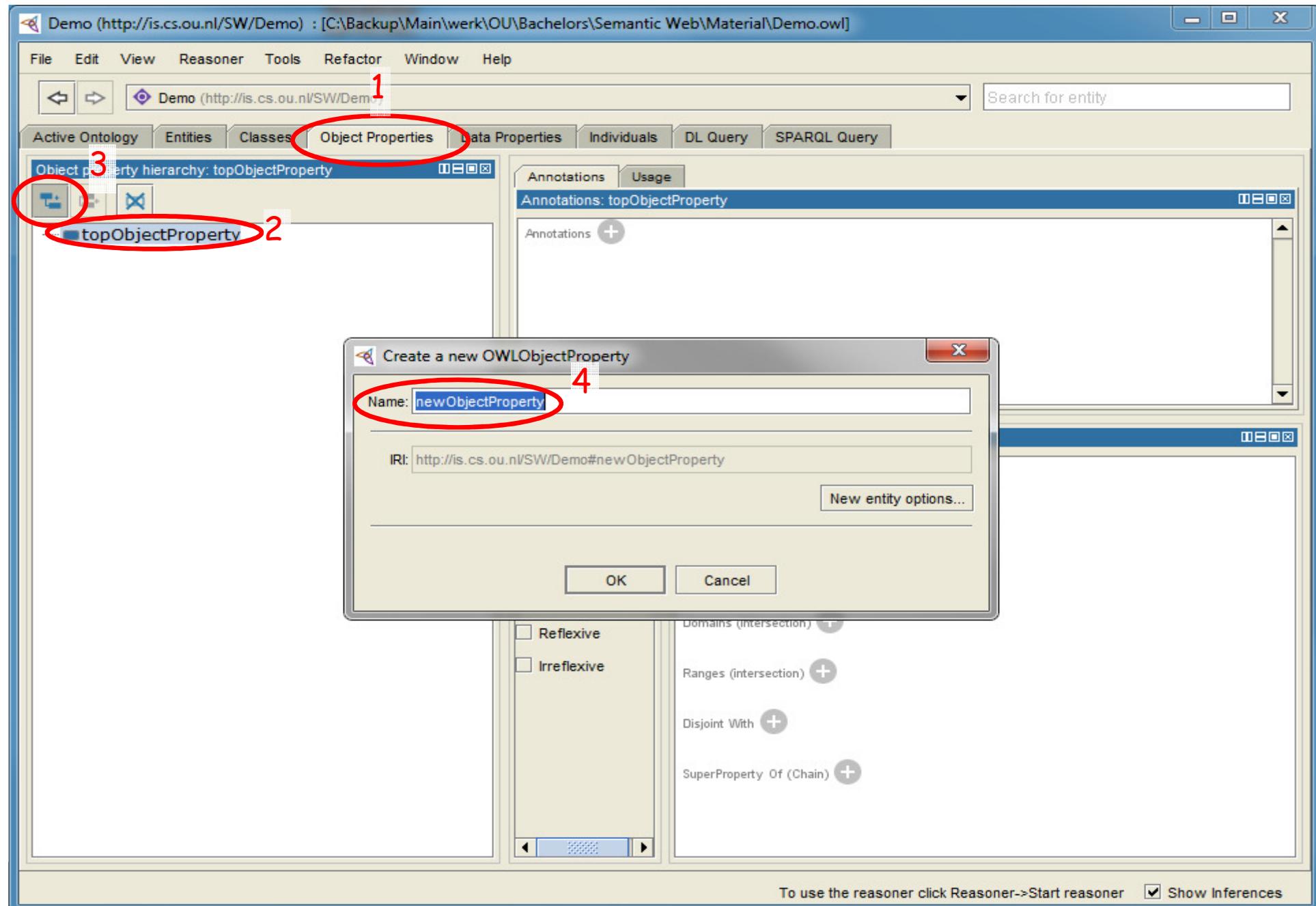
The screenshot shows the RKB Explorer interface for the 'courseware' ontology. The top menu bar includes File, Edit, Ontologies, Reasoner, Tools, Refactor, Tabs, View, Window, and Help. The active tab is 'Reasoner'. The title bar displays the URL [courseware \(http://courseware.rkbexplorer.com/ontologies/courseware\)](http://courseware.rkbexplorer.com/ontologies/courseware).

The left panel shows the 'Asserted class hierarchy' for the 'Course' class. It lists various subclasses of 'Thing' such as 'Assessment-Method', 'Pre-requisite', 'Student-Interaction-Type', 'Composite-Publication', and many others under 'Intangible-Thing' and 'Abstract-Information'. A search bar at the bottom of this panel allows navigating through the hierarchy.

The right panel shows the 'Usage' of the 'Course' class. It lists 23 uses of 'Course' and its properties:

- Course**:
  - subClassOf** Abstract-Information
  - comment** "This class encapsulates a course **that** is taught to students"
  - label** "Course"
- has-assessment-method**:
  - domain** Course
- has-author**:
  - domain** Course
- has-courseware**:
  - domain** Course
- has-infrastructure-requirement**:
  - domain** Course
- has-instructor**:
  - domain** Course
- has-prerequisite**:
  - domain** Course
- has-student-interaction-type**

# Relation = *property*



To use the reasoner click Reasoner->Start reasoner  Show Inferences

# Source = *domain*, Target = *range*

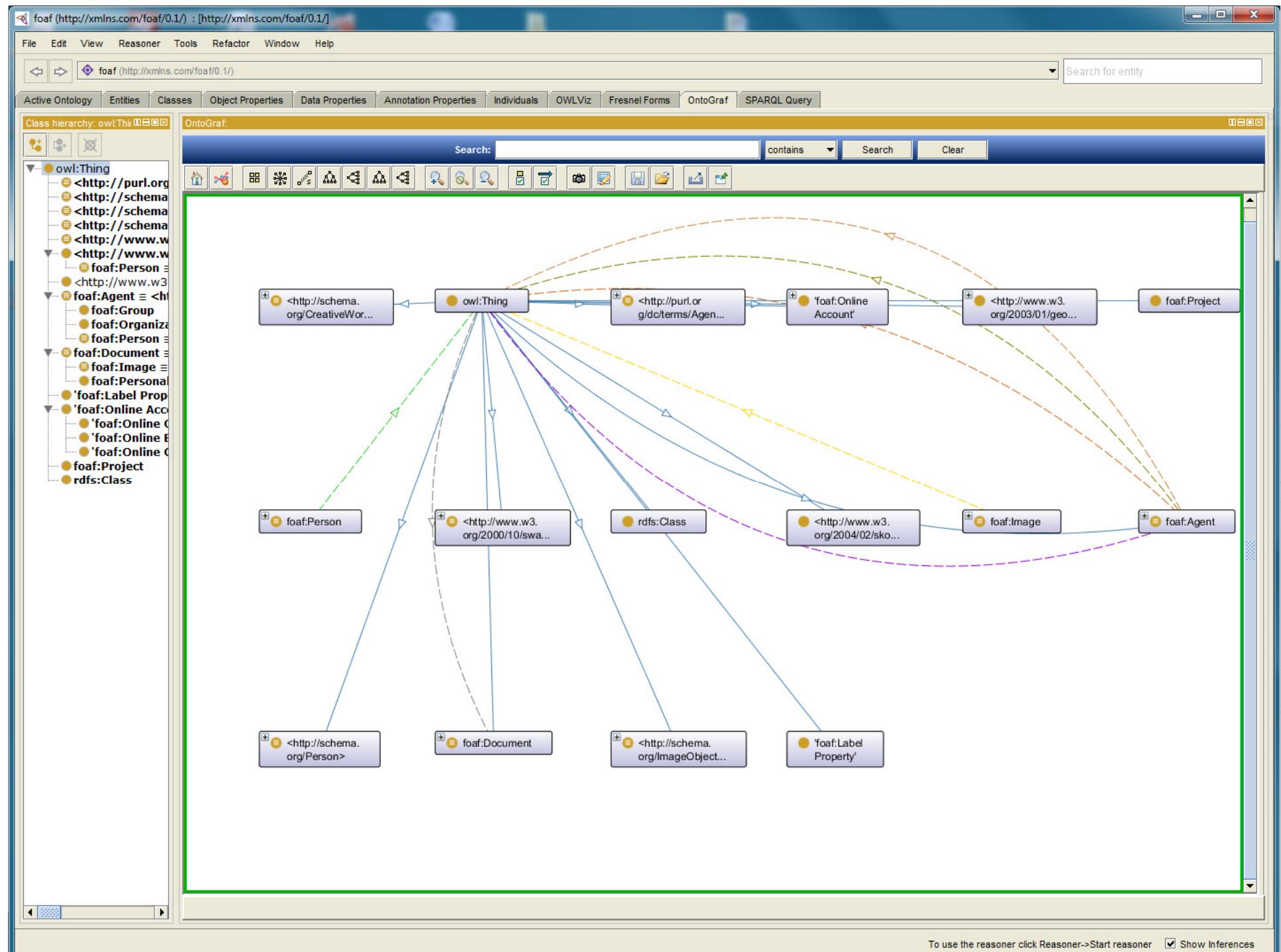
The screenshot shows the Protege 4.3.0 interface with the following annotations:

- Tab 1 (Top Bar):** Demo (http://is.cs.ou.nl/SW/Demo) : [C:\Backup\Main\werk\OU\Bachelors\Semantic Web\Material\Demo.owl]
- Tab 2 (Toolbar):** File, Edit, View, Reasoner, Tools, Refactor, Window, Help
- Tab 3 (Main Navigation):** Active Ontology, Entities, Classes, **Object Properties** (circled in red), Data Properties, Individuals, DL Query, SPARQL Query
- Tab 4 (Left Panel):** Object property hierarchy: newObjectProperty. It shows the hierarchy: topObjectProperty > newObjectProperty. **newObjectProperty** is circled in red and has a red number **2** above it.
- Tab 5 (Annotations Panel):** Annotations tab selected. Shows the annotation: **label** [language: nl] newObjectProperty. A red arrow points from this panel to the **Inverse Of** button in the Characteristic panel.
- Tab 6 (Characteristics Panel):** Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive checkboxes. A red arrow points from the **Inverse Of** button here to the **Inverse Of** button in the Annotations panel.
- Tab 7 (Description Panel):** Domains (intersection), Ranges (intersection), Disjoint With, SuperProperty Of (Chain) buttons. The **Domains (intersection)** button is circled in red and has a red number **3** above it. A red arrow points from the **Inverse Of** button in the Characteristics panel to the **Inverse Of** button here.
- Tab 8 (Bottom Status Bar):** To use the reasoner click Reasoner->Start reasoner,  Show Inferences

**Inverse too!** is written in red text pointing to the **Inverse Of** button in the Characteristics panel.

**newObjectProperty** dialog (bottom-left):

- Class expression editor
- Data restriction creator
- Class hierarchy
- Object restriction creator
- NewClass (highlighted and circled in red, with red number 4 above it)
- OK, Cancel buttons



# Function = *cardinality is 1*

The screenshot shows the SWtutorial ontology editor interface. The active tab is "Classes". A modal dialog box is open for the class `tut:Course`, specifically for creating a "Data restriction creator".

The dialog has several tabs: "Data restriction creator" (selected), "Object restriction creator", and "Class expression editor".

The "Object restriction creator" tab is active, showing:

- "Restricted property": `owl:topObjectProperty`
- "Value": `tut:examiner`

The "Class expression editor" tab shows:

- "Restriction filler": `owl:Thing`

Below the tabs, there is a "Restriction type" section with a dropdown menu set to "Exactly (exact cardinality)" and a "Cardinality" input field set to "1".

At the bottom of the dialog are "OK" and "Cancel" buttons.

On the left side of the main window, the "Class hierarchy (inferred)" panel shows the inheritance path from `owl:Thing` through `tut:Person` to `tut:Course`.

The top navigation bar includes tabs for "Annotation Properties", "Individuals", "OWLviz", "DL Query", "OntoGraf", "SPARQL Query", "Ontology Differences", and "Data Properties".

# Inconsistency on Protégé

The screenshot shows the Protégé ontology editor interface. The main window displays the 'Class hierarchy' tab for the 'tut:Person' class, which is highlighted in orange. The hierarchy tree shows 'owl:Thing' as the root node, with 'tut:Person' and 'tut:Course' as its subclasses. A tooltip 'Description: tut:lloyd' is visible over the 'tut:Person' node. In the top right corner of the main window, there is a red 'X' icon with a curved arrow pointing towards it, indicating an inconsistency.

**Description: tut:lloyd**

Types +

- tut:Course
- tut:Person

Same Individual As +

**Inconsistent ontology explanation**

Show regular justifications    All justifications  
 Show laconic justifications    Limit justifications to 2

Explanation 1    Display laconic explanation

Explanation for: owl:Thing Sub Class Of owl:Nothing

- tut:lloyd **Type** tut:Person
- tut:lloyd **Type** tut:Course
- tut:Course **DisjointWith** tut:Person

OK

Reasoner active but the ontology is inconsistent    Show Inferences



# Horn clauses in Protégé: entry

The screenshot shows the Protégé ontology editor interface. The main window title is "untitled-ontology-77 (http://www.semanticweb.org/lloyd/ontologies/2012/5/7/untitled-ontology-77) : [C:\...]".

The menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The Reasoner menu is open, showing options: Start reasoner (selected), Synchronize reasoner, Explain inconsistent ontology, Configure..., FaCT++, HermiT 1.3.6, Pellet (selected), Pellet (Incremental), and None.

The left sidebar displays the Active Ontology, Ontology header, Ontology Version, and Annotations sections. The Rules section contains a single rule entry:

```
hasBrother(?y, ?z), hasParent(?x, ?y) -> hasUncle(?x, ?z)
```

The right side of the interface features tabs for Data Properties, Annotation Properties, Individuals, and SPARQL Query. A search bar at the top right says "Search for entity".

At the bottom, there is a footer bar with a CC logo, the text "To use the reasoner click Reasoner->Start reasoner", and a checked checkbox for "Show Inferences".

# Horn clauses in Protégé: inferencing

The screenshot shows the Protégé ontology editor interface with the following key elements:

- Top Bar:** File, Edit, View, Reasoner, Tools, Refactor, Window, Help.
- Title Bar:** untitled-ontology-77 (<http://www.semanticweb.org/lloyd/ontologies/2012/5/7/untitled-ontology-77>) : [C:\Backup\Main\werk\OU\B...]
- Left Sidebar:** Active Ontology, Entities, Classes, Object Properties, Data Properties, Annotation Pro...
  - Class hierarchy (inferred):** Shows a tree structure under "Thing".
  - Members list (inferred):** Shows individuals: Amalia, Constantijn, Willem-Alexander.
- Annotations Tab:** Annotations: Amalia
  - Annotations: +
  - label [language: n] Amalia
- Description Tab:** Description: Amalia
  - Types: + Thing ? @ × ○
  - Same Individual As: +
  - Different Individuals: +
- Property assertions Tab:** Property assertions: Amalia
  - Object property assertions: +
    - hasParent Willem-Alexander
    - hasUncle Constantijn
  - Data property assertions: +
  - Negative object property assertions: +
  - Negative data property assertions: +
- Reasoner Panel:** Reasoner active  Show Inferences
- Explanation Panel:** Explanation for Amalia hasUncle Constantijn
  - Show regular justifications (selected)  All justifications
  - Show laconic justifications
  - Limit justifications to 2
  - Explanation 1  Display laconic explanation
    - Explanation for: Amalia hasUncle Constantijn
    - Willem-Alexander hasBrother Constantijn
    - Amalia hasParent Willem-Alexander
    - hasBrother(?y, ?z), hasParent(?x, ?y) -> hasUncle(?x, ?z)

# Data querying in Protégé: SPARQL

The screenshot shows the Protégé SPARQL Query interface. A red number '1' highlights the 'Search for entity' input field. A red number '2' highlights the 'Show Inferences' checkbox at the bottom right. A red number '3' highlights the SPARQL query text. A red number '4' highlights the 'Execute' button. A red number '5' highlights the 'object' column header in the results table.

1

2

3

4

5

SPARQL query:

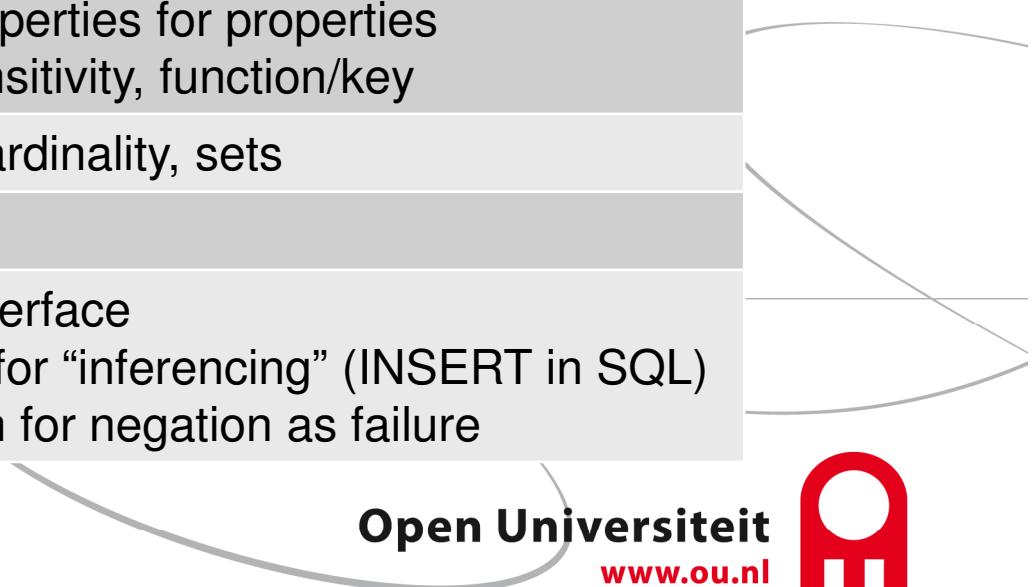
```
SELECT * WHERE { ?subject ?predicate ?object }
```

subject	predicate	object
<http://protege.org/owl2triplestore.owl>	<http://protege.org/owl2triplestore.owl>	<http://is.cs.ou.nl/SWtutorial>
tut:anIndividual	rdf:type	owl:Thing
owl:Thing	rdf:type	owl:Class
tut:anIndividual	rdf:type	owl:NamedIndividual
<http://protege.org/owl2triplestore.owl>	<http://protege.org/owl2triplestore.owl>	"1647802687"^^<http://www.w3.org/
<http://protege.org/owl2triplestore.owl>	<http://protege.org/owl2triplestore.owl>	<http://protege.org/owl2triplestore.owl>
tut:anIndividual	rdf:type	owl:NamedIndividual
<http://protege.org/owl2triplestore.owl>	<http://protege.org/owl2triplestore.owl>	"-219079784"^^<http://www.w3.org/

To use the reasoner click Reasoner->Start reasoner  Show Inferences

# Semantic Web layers

Semantic Web	Logic
RDF	Data: triples, types
RDFS	Description logic <ul style="list-style-type: none"><li>• Class and property hierarchies</li><li>• Domain and range</li></ul>
RDFS-plus	Inferencing properties for properties <ul style="list-style-type: none"><li>• Inverse, transitivity, function/key</li></ul>
OWL	Restrictions, cardinality, sets
SWRL	Horn Clauses
SPARQL	SELECT for interface CONSTRUCT for “inferencing” (INSERT in SQL) Built-in function for negation as failure



# Making a full mapping

Relation Algebra			Semantic Web
	Description	Symb	Context
Ampersand	concept		rdfs:Class
	generalization		subclass
	relation		property
	atom		individual
	complement	-	x -r y distinct class & blank node class w/ restriction?
	intersection	^	x (r ^ s) y * x r y . x s y .
	identity		x   y x a y .
	source, target	*	r :: x * y no cardinality?
	function	=>	r :: x => y r rdfs:domain x; cardinality=1
	univalent	[UNI]	r :: x * y [UNI] rdfs:range y . maxCardinality=1
	total	[TOT]	r :: x * y [TOT] minCardinality=1
	implication	-	r  - s r rdfs:subpropertyOf s .
	equivalence	=	r = s r owl:equivalentProperty s .
	inverse	~	x r~ y r owl:inverseOf r~ .
	composition	;	x (r; s) z * [ s z ; [ owl:inverseOf r ] x . ]
	union/disjunction	U	[ x (r U s) y * x [owl:unionOf ( [owl:oneOf x], [owl:oneOf z] ) . ] y . OPTIONAL {?x teaches OBR.} FILTER(!bound(?x)) <i>Fresnel and Fresnel Forms (OWL Wiki Forms)</i>
	negation as failure		
	interface		

# Beyond the mapping: interface

- Show what you can't conclude
- Query what you can't derive (SPARQL)
- Process rules
  - Just tell the user a rule is broken
  - Don't conclude it's broken because then it's harder to un-conclude
- Prototype system interface ...

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



# Stap 5 - 1

Main Mijn VOG Aanvragen Overzicht (COVOG) Overzicht (Gemeente) Overzicht (Organisaties)

Save Cancel

**Aanvraagnummer**  
NV-VOG-123

**VOG aanvraag**

Doel van de aanvraag  
Acme B.V. : Financieel medewerker

Soort identiteitsbewijs (paspoort, rijbewijs)  
Rijbewijs

Serienummer  
RDW678214

Komen deze gegevens overeen met de persoon die u voor zich heeft?  
Ja

DigiD

Wachtwoord

GBA gegevens van aanvrager

**Behandelend ambtenaar**  
R. Oden (Noordenveld)

**Bijzonderheden**

Bijzonderheden geconstateerd (Ja/Nee)?  
Nee

Ja, toelichting

**Persisteren**

Persisteren in de aanvraag (Ja/Nee)?  
Nee

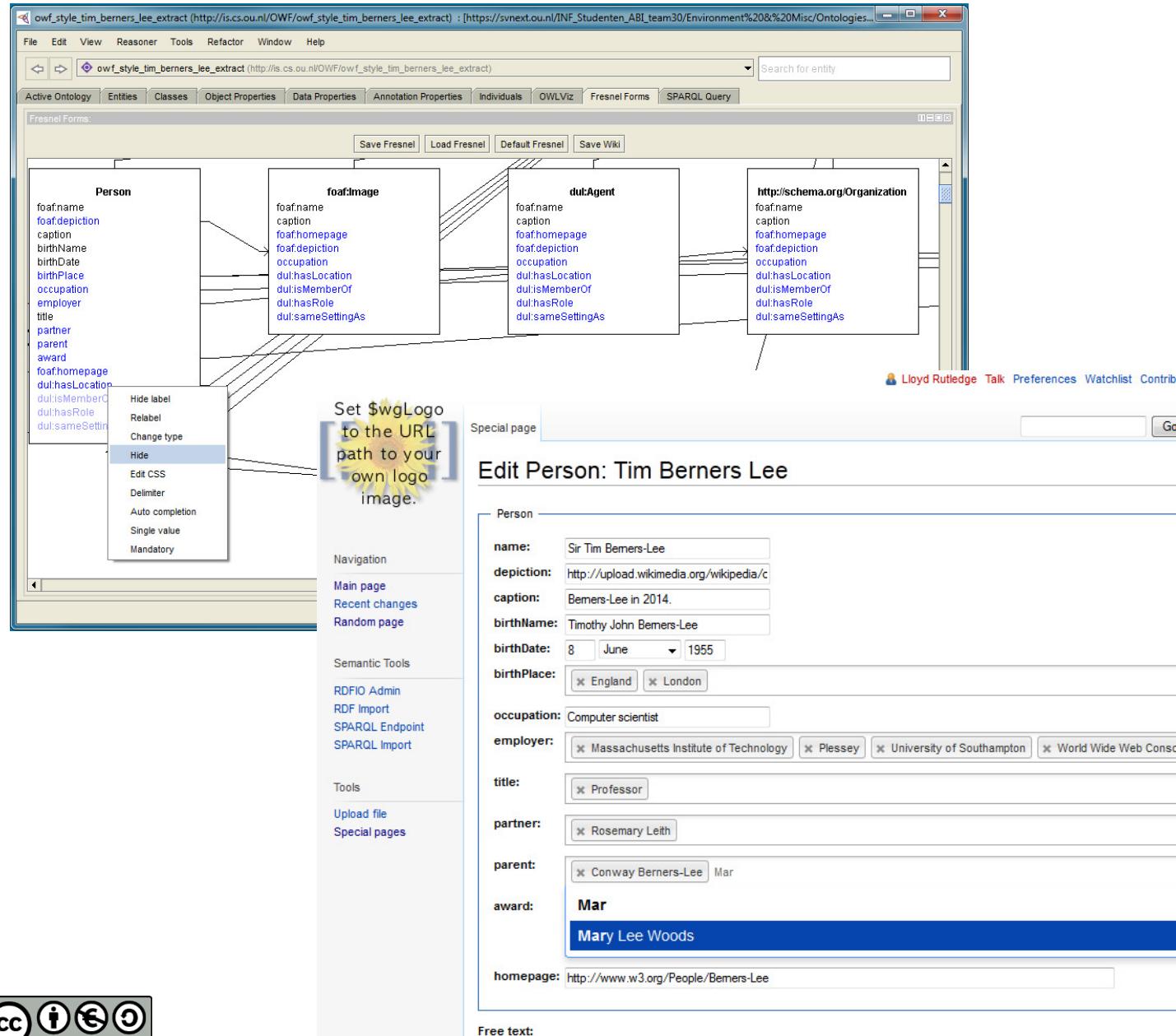
Ja, toelichting

**COVOG advies**

COVOG om advies gevraagd omtrent ontvankelijkheid (J)  
Nee



# OU's Fresnel Forms plugin for Protégé



# Fresnel Form rules enforced by interface

- Property to box assignment from source/domain
- Autocompletion from target/range
- Mandatory from function and total / cardinality  $\geq 1$
- No value list if univalent / maximum cardinality =1
- Room for much more!
  - Semantic MediaWiki and extensions offer much functionality

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



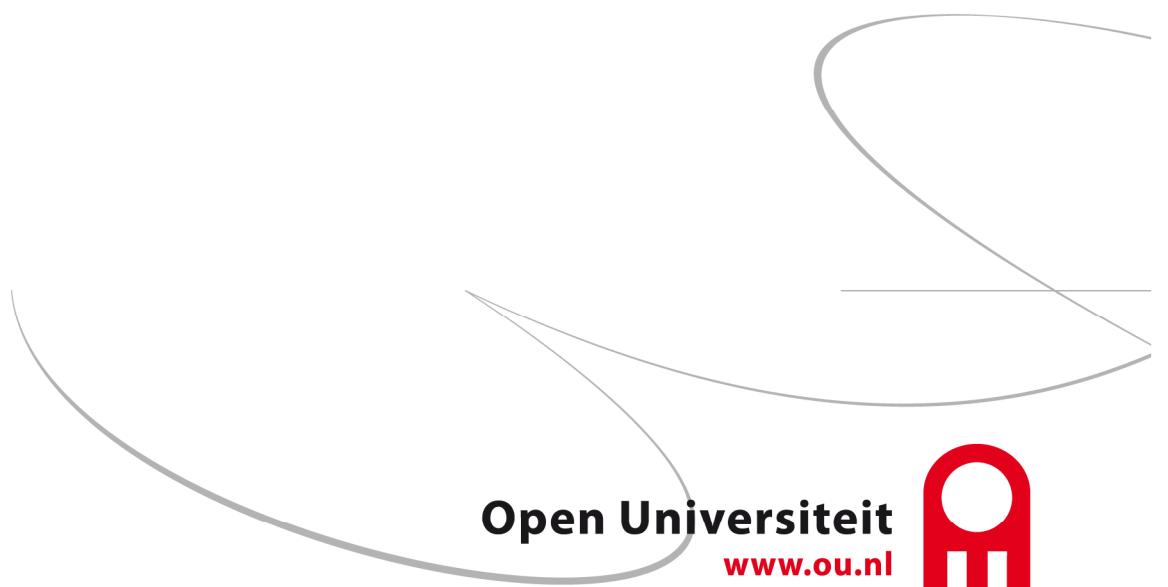
# Future work

- Education
  - Semantic Web variant of the OU Rule-based Design course
  - More emphasis on logical layers in OU Semantic Web course
- Research
  - More with quick prototyping on wikis
  - Rule patterns and ontology patterns
  - Scholarly comparison of logical formalisms
- Tool development
  - Make Semantic MediaWiki function like Ampersand's wiki
  - Relation algebra rule editor on Protégé

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



# Questions?



**Open Universiteit**  
[www.ou.nl](http://www.ou.nl)

