MANCHESTER 1824

The University of Manchester

Explanation in Ontology Languages

Bijan Parsia, Thomas Schneider
University of Manchester

ESSLLI 2009, 27-31 July, Bordeaux

With thanks to Matthew Horridge and Johannes Bauer

Outline

Mon	Introduction
Tue	Computation of justifications
Wed	Fine-grained justifications
Thu	Lemmata
Fri	Model based explanations

Monday

Introduction

Schedule for today

- Ontologies and ontology languages
- Background on explanation
- Introduction to justifications and associated services

Ontologies and ontology languages

Ontologies

- Are logical theories
- Capture a domain of interest
- Contain explicit knowledge
- Implicit knowledge can be inferred
- This is possible through the formal semantics underlying ontology languages

Ontology languages

- Often based on Description Logics (DLs)
  - Class/object/role paradigm
  - DLs have clear, formal semantics
  - (Decidable) fragments of first order logic
  - Standard, implemented inference services

Ontology languages

- Often based on Description Logics (DLs)
  - Class/object/role paradigm
  - DLs have clear, formal semantics
  - (Decidable) fragments of first order logic
  - Standard, implemented inference services

Many services are "automagic" (e.g., classification)

DL examples

Women are female persons.

Every mother is a female parent.

Every parent is a person.

Every parent has at least one child, each of which is a person.

DL examples

Women are female persons.

$\text{Woman} \equiv \text{Female} \sqcap \text{Person}$

Every mother is a female parent.

Every parent is a person.

Every parent has at least one child, each of which is a person.

DL examples

Women are female persons.

$\mathsf{Woman} \equiv \mathsf{Female} \sqcap \mathsf{Person}$

Every mother is a female parent.

$\mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \mathsf{Parent}$

Every parent is a person.

Every parent has at least one child, each of which is a person.

DL examples

Women are female persons.

$\mathsf{Woman} \equiv \mathsf{Female} \sqcap \mathsf{Person}$

Every mother is a female parent.

$\mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \mathsf{Parent}$

Every parent is a person.

$\mathsf{Parent} \sqsubseteq \mathsf{Person}$

Every parent has at least one child, each of which is a person.

DL examples

Women are female persons.

$\mathsf{Woman} \equiv \mathsf{Female} \sqcap \mathsf{Person}$

Every mother is a female parent.

$\mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \mathsf{Parent}$

Every parent is a person.

$\mathsf{Parent} \sqsubseteq \mathsf{Person}$

Every parent has at least one child, each of which is a person.

$\mathsf{Parent} \sqsubseteq \geqslant 1\mathsf{hasChild} \sqcap \forall \mathsf{hasChild}.\mathsf{Person}$

DL examples

Women are female persons.

$\mathsf{Woman} \equiv \mathsf{Female} \sqcap \mathsf{Person}$

Every mother is a female parent.

$\mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \mathsf{Parent}$

Every parent is a person.

$\mathsf{Parent} \sqsubseteq \mathsf{Person}$

implies:

Every mother is a woman.

$\mathsf{Mother} \sqsubseteq \mathsf{Woman}$

Every parent has at least one child, each of which is a person.

$\mathsf{Parent} \sqsubseteq \geqslant 1\mathsf{hasChild} \sqcap \forall \mathsf{hasChild}.\mathsf{Person}$

DL examples

$$\mathsf{Parent} \sqsubseteq \geqslant 1\mathsf{hasChild} \sqcap \forall \mathsf{hasChild}.\mathsf{Person}$$

DL examples

$$\mathsf{Parent} \sqsubseteq \geqslant 1\, \mathsf{hasChild} \sqcap \forall\, \mathsf{hasChild}.\mathsf{Person}$$

DL examples

$$\mathsf{Parent} \sqsubseteq \geqslant 1\, \mathsf{hasChild} \sqcap \forall\, \mathsf{hasChild}.\mathsf{Person}$$

class name
(concept name)

class name
(concept name)

DL examples

$$\mathsf{Parent} \sqsubseteq \geqslant 1\, \mathsf{hasChild} \sqcap \forall\, \mathsf{hasChild}.\mathsf{Person}$$

class name
(concept name)

property name
(role name)

property name
(role name)

class name
(concept name)

DL examples

$$\text{Parent} \sqsubseteq \geqslant 1\,\text{hasChild} \sqcap \forall\,\text{hasChild}.\text{Person}$$

class name
(concept name)

property name
(role name)

logical connectives

property name
(role name)

class name
(concept name)

DL examples

axiom

$$\text{Parent} \sqsubseteq \geqslant 1\ \text{hasChild} \sqcap \forall\, \text{hasChild}.\text{Person}$$

class name (concept name)

property name (role name)

logical connectives

property name (role name)

class name (concept name)

DL syntax

- class names; property names:

  $A, B, \ldots ; \quad P, Q, \ldots$

- complex classes:

  $$\top, \ A, \ \neg C, \ C \sqcap D, \ \exists R.C, \ \geqslant n\, R.C$$

  A atomic; C, D possibly complex;
  R atomic property or inverse (P^-); $n \in \mathbb{N}$

- convention: A, B atomic classes; $C, D, \ldots$ arbitrary

- syntactic sugar:

  $$\bot = \neg\top \qquad \forall R.C = \neg\exists R.\neg C$$

  $$C \sqcup D = \neg C \sqcap \neg D \qquad \leqslant n\, R.C = \neg(\geqslant n{+}1\, R.C)$$

DL syntax

$C \sqsubseteq D$	subClassOf axiom
$C \equiv D$	shortcut for $C \sqsubseteq D$ and $D \sqsubseteq C$
$R_1 \circ \cdots \circ R_n \sqsubseteq S$	subPropertyChain axiom

+ more features, mostly syntactic sugar:
disjointness, transitivity, reflexivity, ...

DL syntax vs. OWL Manchester syntax

DL	*OWL Manchester syntax*
concept, role	class, property
$\top, \bot$	`Thing`, `Nothing`
$\neg C$	`not` C
$C \sqcap D$, $C \sqcup D$	C `and` D, C `or` D
$\exists R.C$, $\forall R.C$	R `some` C, R `only` C
$\geqslant n\, R.C$, $\leqslant n\, R.C$	R `min` n C, R `max` n C
$=n\, R.C$	R `exactly` n C

DL semantics

- Interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ $\quad \Delta^{\mathcal{I}} \neq \emptyset$!
  with $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for class names A
  and $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for property names P

- Inductive transfer to arbitrary props / classes

$$
\begin{aligned}
(P^{-})^{\mathcal{I}} &= \{(x, y) \mid (y, x) \in P^{\mathcal{I}}\} \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{for some } y \in C^{\mathcal{I}},\ (x, y) \in R^{\mathcal{I}}\} \\
(\geqslant n\, R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \#\{y \in C^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\} \geqslant n\}
\end{aligned}
$$

DL semantics

Interpretation $\mathcal{I}$ satisfies ...

- axiom $A \sqsubseteq B$ if $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$
- axiom $R_1 \circ \cdots \circ R_n \sqsubseteq S$ if:

  whenever $xR_1^{\mathcal{I}} z_1 R_2^{\mathcal{I}} z_2 R_3^{\mathcal{I}} \ldots R_{n-1}^{\mathcal{I}} z_{n-1} R_n^{\mathcal{I}} y$,
  then $xS^{\mathcal{I}} y$

We write $\mathcal{I} \models \alpha$.

We write $\mathcal{I} \models \mathcal{O}$ if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{O}$.

DL semantics

- Interpretation $\mathcal{I}$ is a *model* of ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies every axiom in $\mathcal{O}$.

  We write $\mathcal{I} \models \mathcal{O}$.

- Ontology $\mathcal{O}$ is *inconsistent* if it has no model.
- Class C is *satisfiable* if $C^{\mathcal{I}} \neq \emptyset$ for every interpretation $\mathcal{I}$.
- Axiom α is *entailed* by $\mathcal{O}$ $\quad (\mathcal{O} \models \alpha)$ if for all $\mathcal{I}$: $\quad \mathcal{I} \models \mathcal{O} \;\Rightarrow\; \mathcal{I} \models \alpha$

Ontologies

(Bio-)Medicine

Chemistry

... are used
in these fields
and others

Astronautics

Linguistics

Description of
business processes

(Web) Service
description

MANCHESTER 1824

The University of Manchester

Ontologies

File Edit Reasoner Tools Refactor Tabs View Window Help

tambis-full.owl http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/tambis-full.owl

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLViz | DL Query

Asserted Class Hierarchy: collection-book

- name
  - author-name
  - editor-name
  - gene-name
  - journal-name
  - protein-name
- work-title
  - book-title
- mental-organisation
  - biological-classification
    - organism-classification
      - eukaryote
      - prokaryote
      - species
- work
  - published-material
    - authored-chapter
    - book
    - collection-book
    - journal

Asserted class hierarchy | Inferred class hierarchy

Object Properties:

- published-by
- has-charge
- transforms
- has-unit
- ec-number-of
- atom-type
- has-function
- has-organism-classification
- codes-for
- catalysed-by
- expressed-by

Object property hierarchy | Data property hierarchy | Individuals

Selected entity:

Class Annotations: collection-book

Annotations

Class Annotations | Class Usage

Class Description: collection-book

Equivalent classes

Superclasses

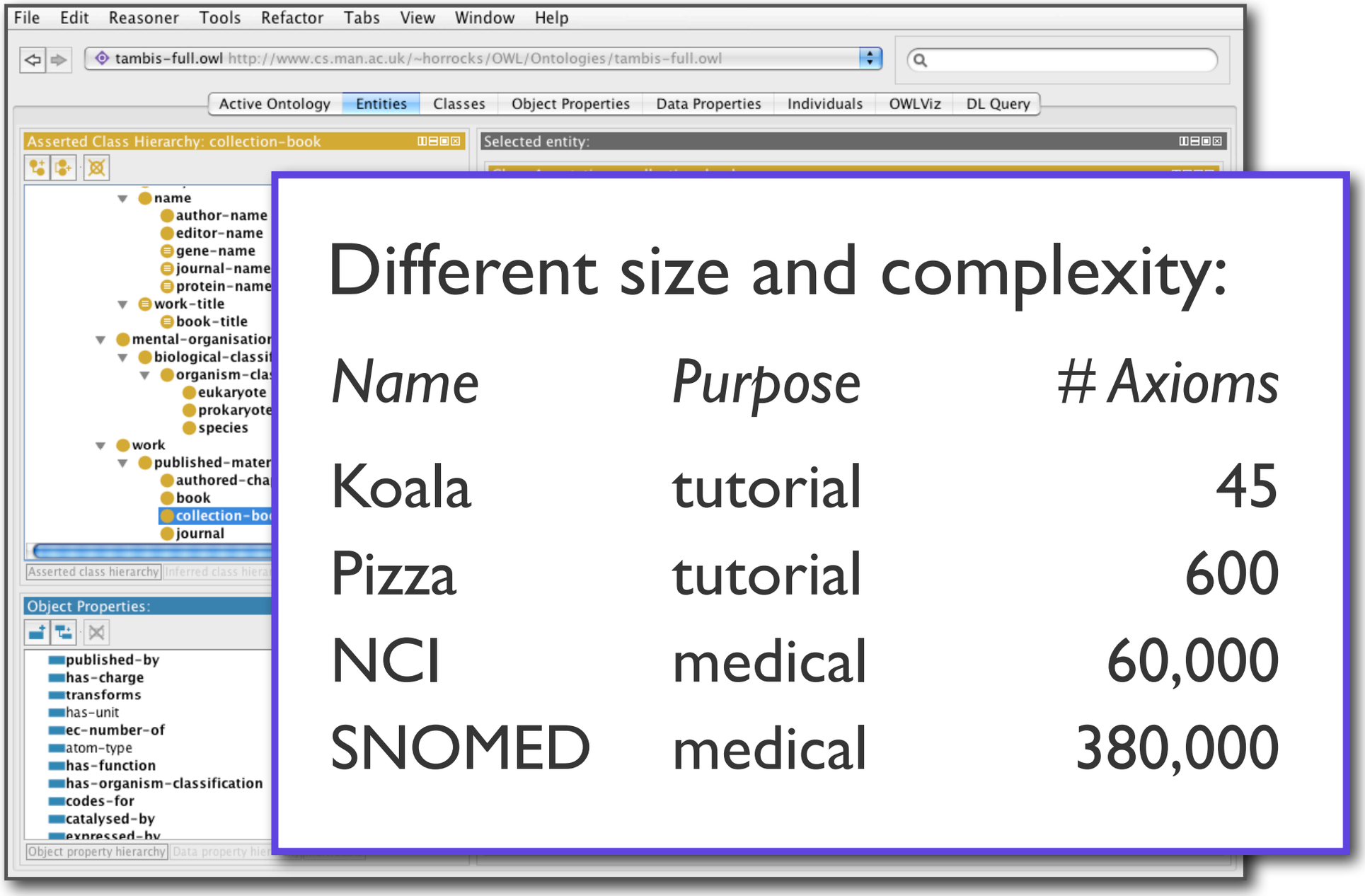
- published-material
- has-editor some editor-name
- has-part some authored-chapter
- has-part min 1 Thing

Inherited anonymous classes

- published-by some publishing-institution
- has-title some work-title
- has-publication-year some year
- published-by only publishing-institution
- structure and mental

Instances

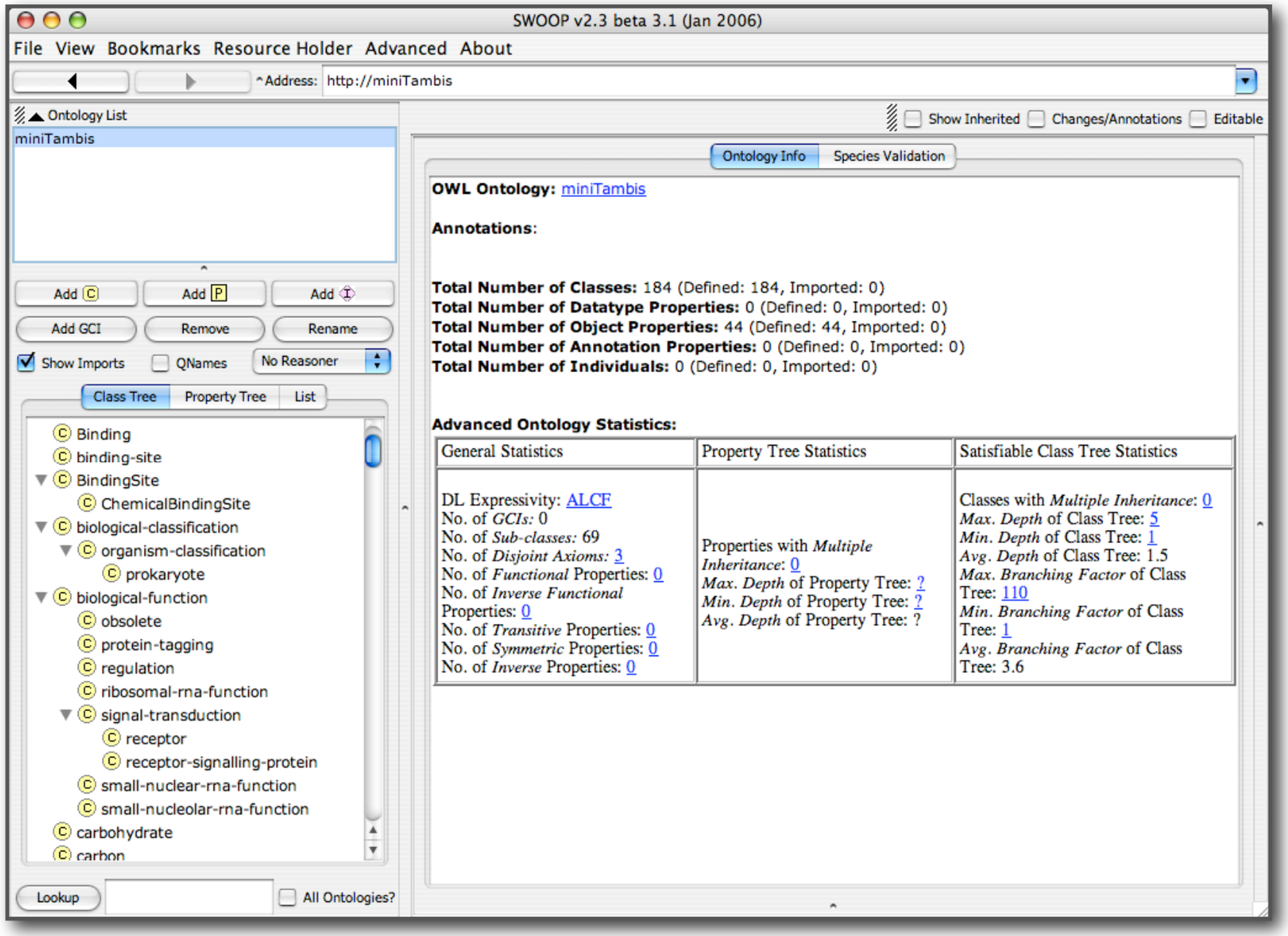
Ontologies



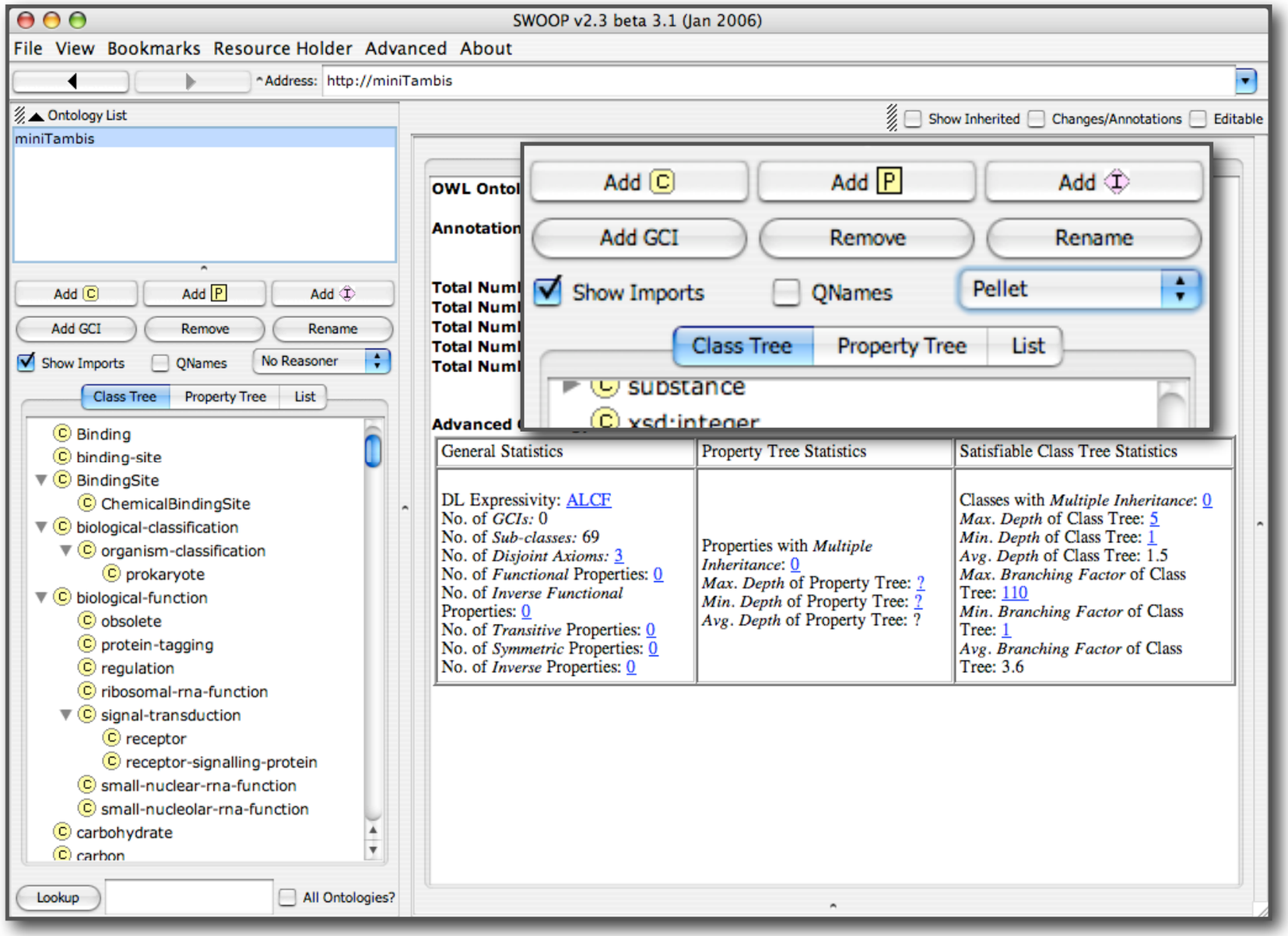
Different size and complexity:

Name	*Purpose*	*# Axioms*
Koala	tutorial	45
Pizza	tutorial	600
NCI	medical	60,000
SNOMED	medical	380,000

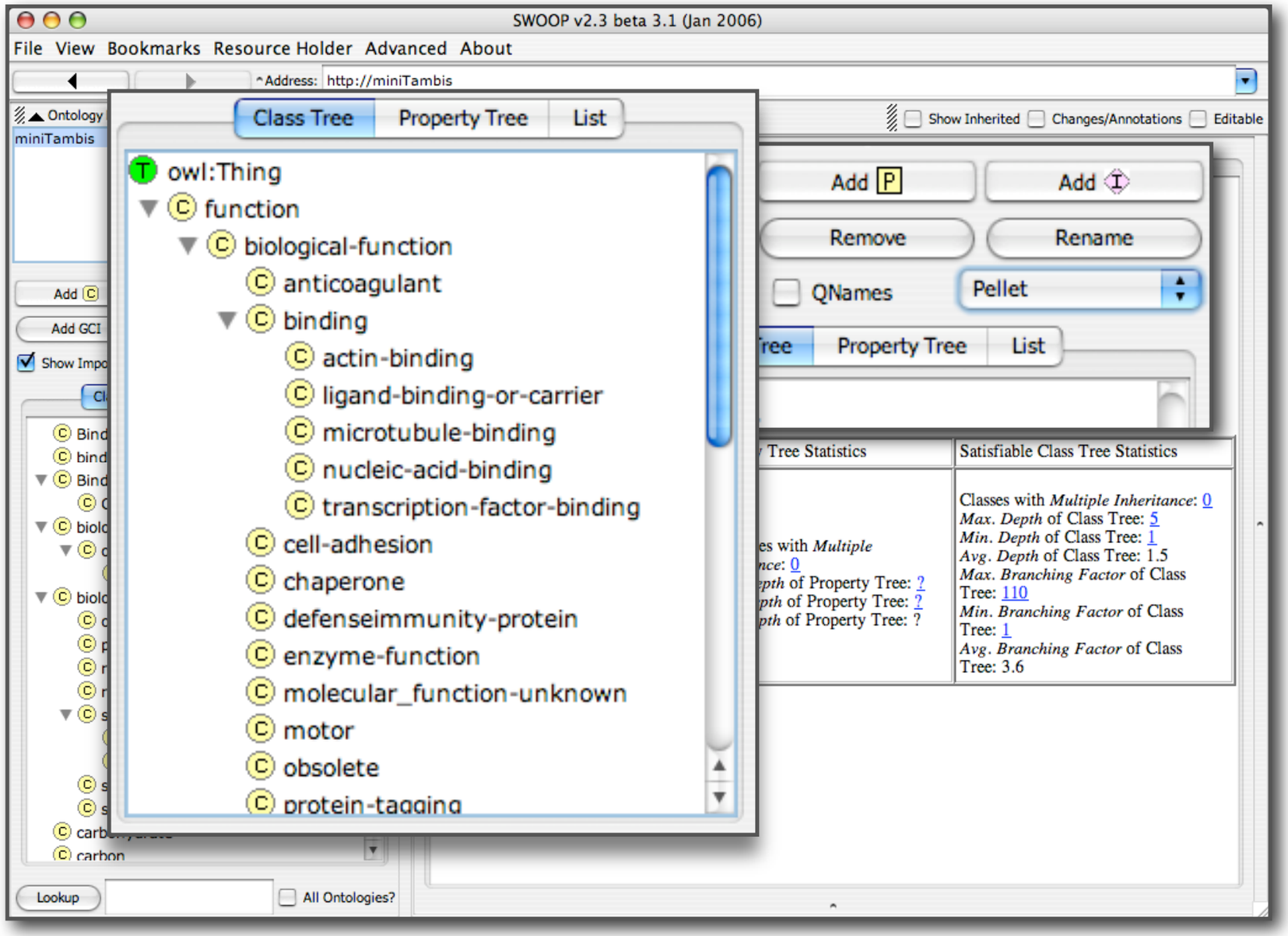
Ontologies


SWOOP v2.3 beta 3.1 (Jan 2006)
File View Bookmarks Resource Holder Advanced About
Address: http://miniTambis
Ontology List
miniTambis
Show Inherited
Changes/Annotations
Editable
Ontology Info
Species Validation
OWL Ontology: miniTambis
Annotations:
Total Number of Classes: 184 (Defined: 184, Imported: 0)
Total Number of Datatype Properties: 0 (Defined: 0, Imported: 0)
Total Number of Object Properties: 44 (Defined: 44, Imported: 0)
Total Number of Annotation Properties: 0 (Defined: 0, Imported: 0)
Total Number of Individuals: 0 (Defined: 0, Imported: 0)
Advanced Ontology Statistics:
General Statistics
Property Tree Statistics
Satisfiable Class Tree Statistics
DL Expressivity: ALCF
No. of GCIs: 0
No. of Sub-classes: 69
No. of Disjoint Axioms: 3
No. of Functional Properties: 0
No. of Inverse Functional Properties: 0
No. of Transitive Properties: 0
No. of Symmetric Properties: 0
No. of Inverse Properties: 0
Properties with Multiple Inheritance: 0
Max. Depth of Property Tree: ?
Min. Depth of Property Tree: ?
Avg. Depth of Property Tree: ?
Classes with Multiple Inheritance: 0
Max. Depth of Class Tree: 5
Min. Depth of Class Tree: 1
Avg. Depth of Class Tree: 1.5
Max. Branching Factor of Class Tree: 110
Min. Branching Factor of Class Tree: 1
Avg. Branching Factor of Class Tree: 3.6
Add C
Add P
Add I
Add GCI
Remove
Rename
Show Imports
QNames
No Reasoner
Class Tree
Property Tree
List
Binding
binding-site
BindingSite
ChemicalBindingSite
biological-classification
organism-classification
prokaryote
biological-function
obsolete
protein-tagging
regulation
ribosomal-rna-function
signal-transduction
receptor
receptor-signalling-protein
small-nuclear-rna-function
small-nucleolar-rna-function
carbohydrate
carbon
Lookup
All Ontologies?

Ontologies


SWOOP v2.3 beta 3.1 (Jan 2006)
File View Bookmarks Resource Holder Advanced About
Address: http://miniTambis
Ontology List
miniTambis
Show Inherited
Changes/Annotations
Editable
Add C
Add P
Add I
Add GCI
Remove
Rename
Show Imports
QNames
Pellet
No Reasoner
Class Tree
Property Tree
List
substance
xsd:integer
Binding
binding-site
BindingSite
ChemicalBindingSite
biological-classification
organism-classification
prokaryote
biological-function
obsolete
protein-tagging
regulation
ribosomal-rna-function
signal-transduction
receptor
receptor-signalling-protein
small-nuclear-rna-function
small-nucleolar-rna-function
carbohydrate
carbon
Lookup
All Ontologies?
General Statistics
Property Tree Statistics
Satisfiable Class Tree Statistics
DL Expressivity: ALCF
No. of GCIs: 0
No. of Sub-classes: 69
No. of Disjoint Axioms: 3
No. of Functional Properties: 0
No. of Inverse Functional Properties: 0
No. of Transitive Properties: 0
No. of Symmetric Properties: 0
No. of Inverse Properties: 0
Properties with Multiple Inheritance: 0
Max. Depth of Property Tree: ?
Min. Depth of Property Tree: ?
Avg. Depth of Property Tree: ?
Classes with Multiple Inheritance: 0
Max. Depth of Class Tree: 5
Min. Depth of Class Tree: 1
Avg. Depth of Class Tree: 1.5
Max. Branching Factor of Class Tree: 110
Min. Branching Factor of Class Tree: 1
Avg. Branching Factor of Class Tree: 3.6

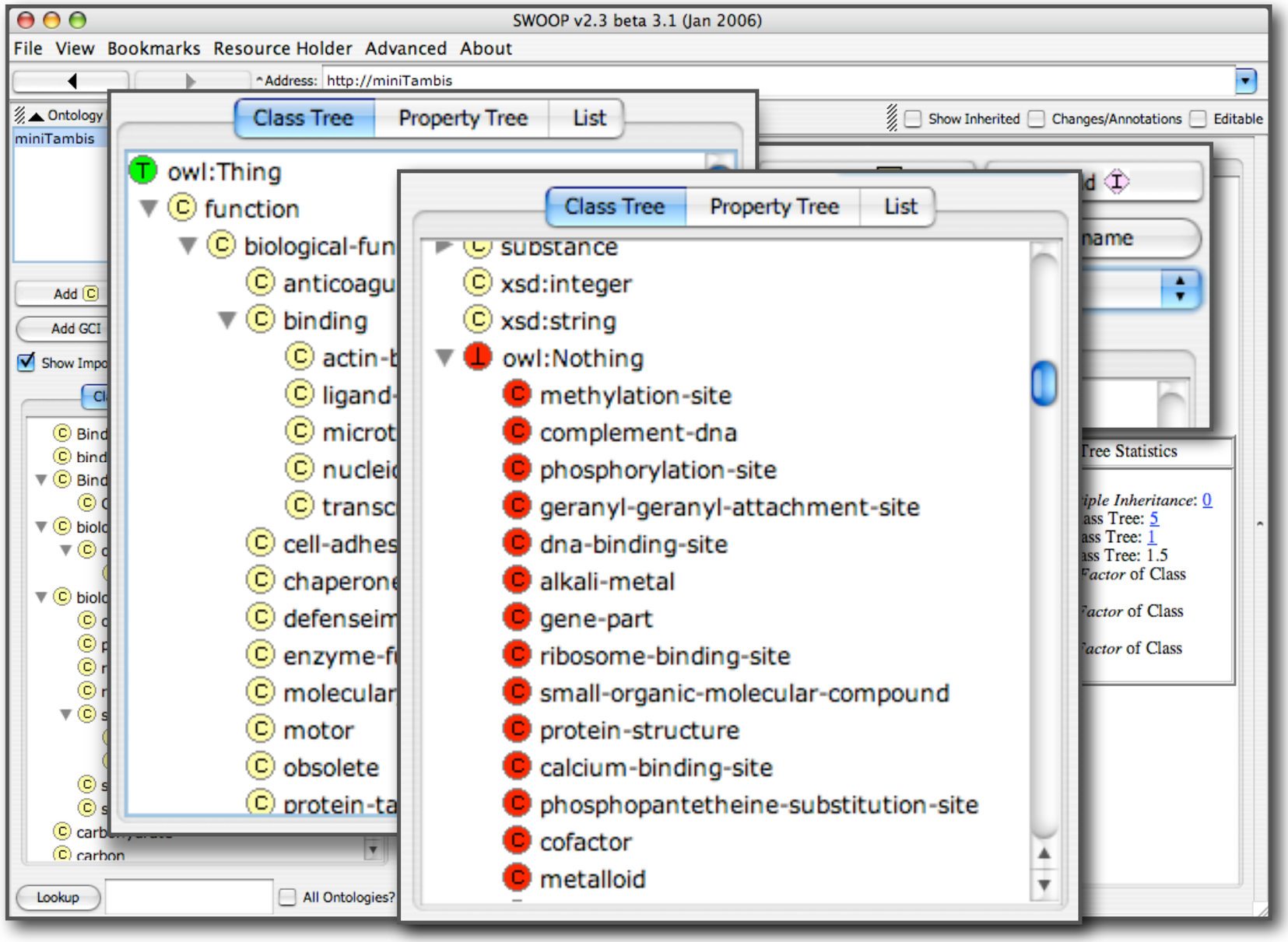
Ontologies


SWOOP v2.3 beta 3.1 (Jan 2006)
File View Bookmarks Resource Holder Advanced About
Address: http://miniTambis
Class Tree
Property Tree
List
owl:Thing
function
biological-function
anticoagulant
binding
actin-binding
ligand-binding-or-carrier
microtubule-binding
nucleic-acid-binding
transcription-factor-binding
cell-adhesion
chaperone
defenseimmunity-protein
enzyme-function
molecular_function-unknown
motor
obsolete
protein-tagging
Add P
Add I
Remove
Rename
QNames
Pellet
Show Inherited
Changes/Annotations
Editable
Satisfiable Class Tree Statistics
Classes with Multiple Inheritance: 0
Max. Depth of Class Tree: 5
Min. Depth of Class Tree: 1
Avg. Depth of Class Tree: 1.5
Max. Branching Factor of Class Tree: 110
Min. Branching Factor of Class Tree: 1
Avg. Branching Factor of Class Tree: 3.6
Lookup
All Ontologies?

MANCHESTER 1824

The University of Manchester

Ontologies



"Large" User Base

- Large and Growing
  - (126,870 registered Protégé Users)
  - (Standardization helps)
  - Bio-medical applications huge
- Useful for research!
  - Diverse users, tasks, issues
  - High impact

Reasoning services

Name	*Question*
Consistency checking	$\mathcal{O} \models \top \sqsubseteq \bot$?
Subsumption checking	$\mathcal{O} \models C \sqsubseteq D$?
Class hierarchy computation	$\mathcal{O} \models A \sqsubseteq B$?
Unsatisfiability test	$\mathcal{O} \models C \sqsubseteq \bot$?
Instance checking	$\mathcal{O} \models a : C$?

Reasoning services ...

- ... are interreducible, e.g.:

$$\mathcal{O} \models \top \sqsubseteq \bot \quad \Longleftrightarrow \quad \mathcal{O} \models A \sqcup \neg A \sqsubseteq \bot$$

$$\mathcal{O} \models C \sqsubseteq D \quad \Longleftrightarrow \quad \mathcal{O} \models C \sqcap \neg D \sqsubseteq \bot$$

~> Let's focus on unsatisfiability!

- ... do not provide explanations

Need for explanations

- Some entailments undesirable (e.g., unsatisfiable classes)
- Debugging necessary
  ~> delete/repair axioms responsible,
  but which axioms?
- Needle(s) in the haystack (of a large ontology)

Background on explanation

Context!

	Time	
	Development	Deployment
Ontology Developer		
App Programmer		
App user		

Context!

	Time	
	Development	Deployment
Ontology Developer	Our focus	
App Programmer		
App user		

OntEng Tasks

Understanding entailments

Debugging and repair

Understanding justifications

Ontology comprehension

OntEng Tasks

Understanding entailments

Debugging and repair

Understanding justifications

Ontology comprehension

Understanding entailments

- User notices an entailment.
- Decides to obtain an explanation for it in order to find out why it holds.

Understanding justifications

- Justification for an entailment in an ontology has been obtained.
- User wants to understand the justification better.

Ontology comprehension

- User is faced with a new ontology.
- Wants to get a better picture of it.
- Among the possible metrics:
  - ➡ Number of entailments
  - ➡ Average number of justifications for an entailment
  - ➡ ...

Debugging and repair

- User is faced with
  1. an inconsistent ontology
  2. an unsatisfiable class
  3. or "some" undesired entailment.
- Determine the cause and debug.

Debugging and repair

- User is faced with
  1. an inconsistent ontology
  2. an unsatisfiable class
  3. or "some" undesired entailment.
- Determine the cause and debug.

A lot of research on 1 & 2 can be done without domain knowledge

Debugging and repair

- User is faced with
  1. an inconsistent ontology
  2. an unsatisfiable class
  3. or "some" undesired entailment.
- Determine the cause and debug.

A lot of research on 1 & 2 can be done without domain knowledge

1 & 2 have high, visible user value

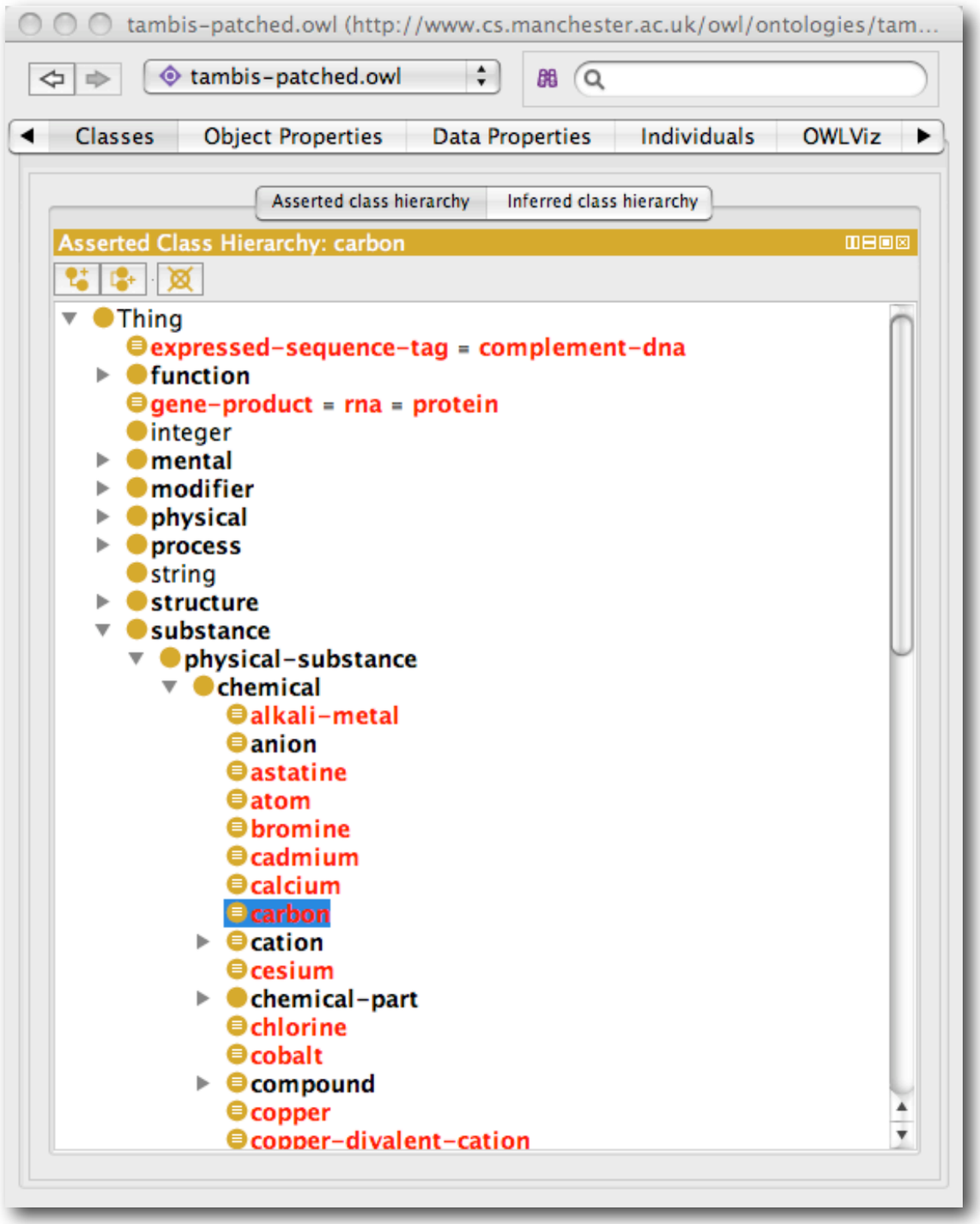
Manual Debugging

Life without explanations

- Open and classify the TAMBIS ontology `tambis-patched.owl` in Protégé
- Open the "Asserted class hierarchy" and expand the following terms:

  substance
  physical-substance
  chemical

Life without explanations



Oops!

144 out of 395 classes are unsatisfiable (red).

Why?

Why are my classes unsatisfiable?

- What have we said in the ontology that **causes** these classes to be unsat?
- What do we have to change to **repair** it?
- Where do we **start**?

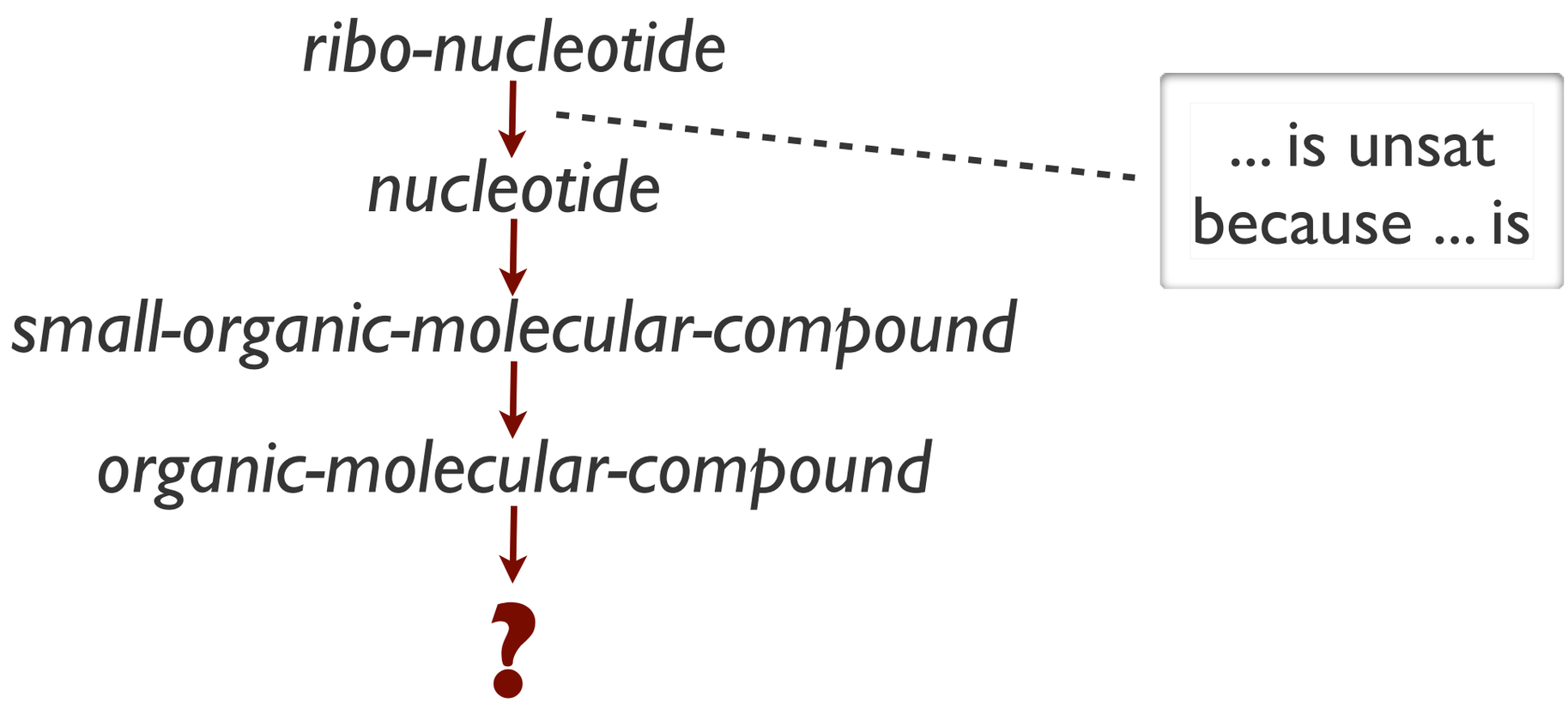
Where do we start? Tracing

- Open and classify the TAMBIS ontology `tambis-patched.owl` in Protégé
- Open the "Asserted class hierarchy" and expand the following terms:

  substance
  physical-substance
  chemical
  organic-molecular-compound
  small-organic-molecular-compound
  nucleotide
  ribo-nucleotide

Where do we start? Tracing

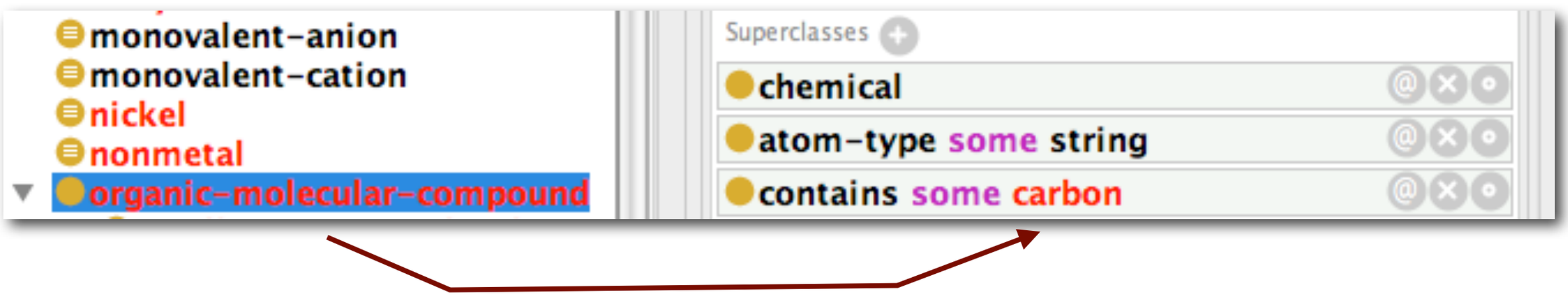
Navigate through the subclass hierarchy.
If a class is unsat, then so are its subclasses.



Tracing in the old days

The only way was to manually follow ...

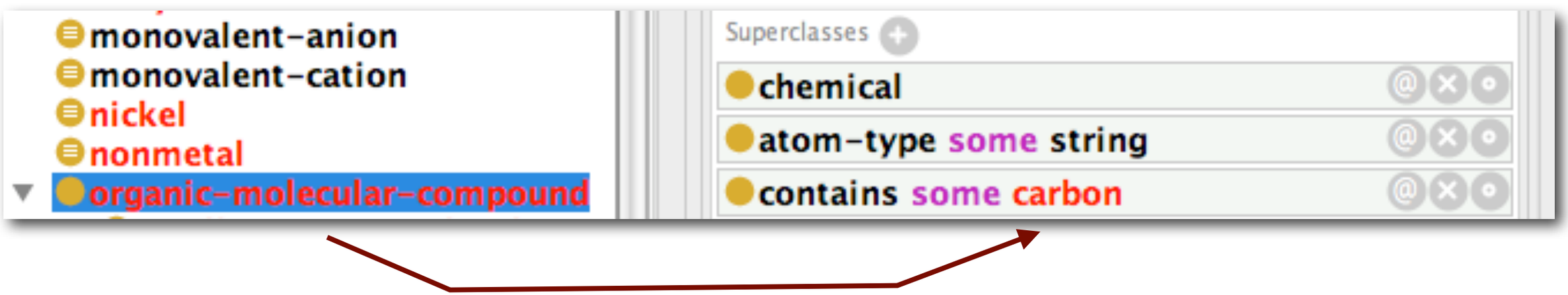
- unsatisfiable named superclasses
- or unsatisfiable fillers of restrictions in anonymous superclasses:



Tracing in the old days

The only way was to manually follow ...

- unsatisfiable named superclasses
- or unsatisfiable fillers of restrictions in anonymous superclasses:

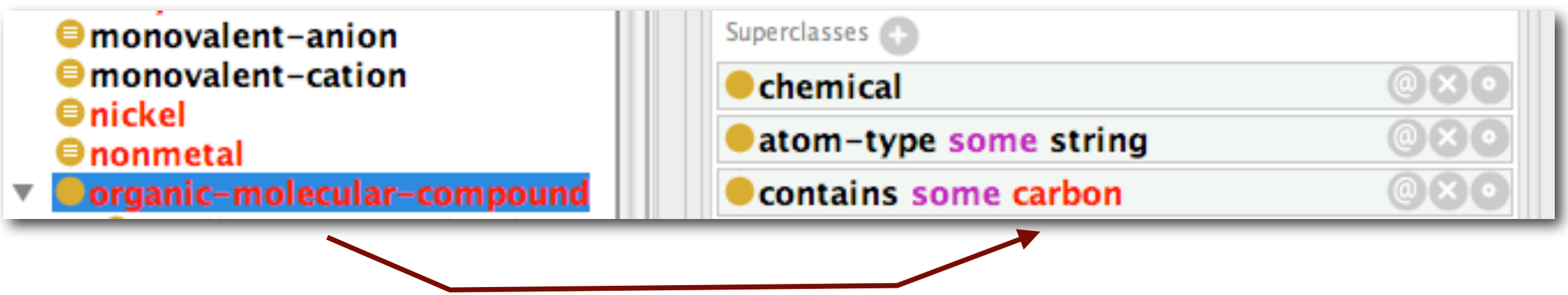


Tedious for big ontologies! ☹

Tracing in the old days

The only way was to manually follow ...

- unsatisfiable named superclasses
- or unsatisfiable fillers of restrictions in anonymous superclasses:



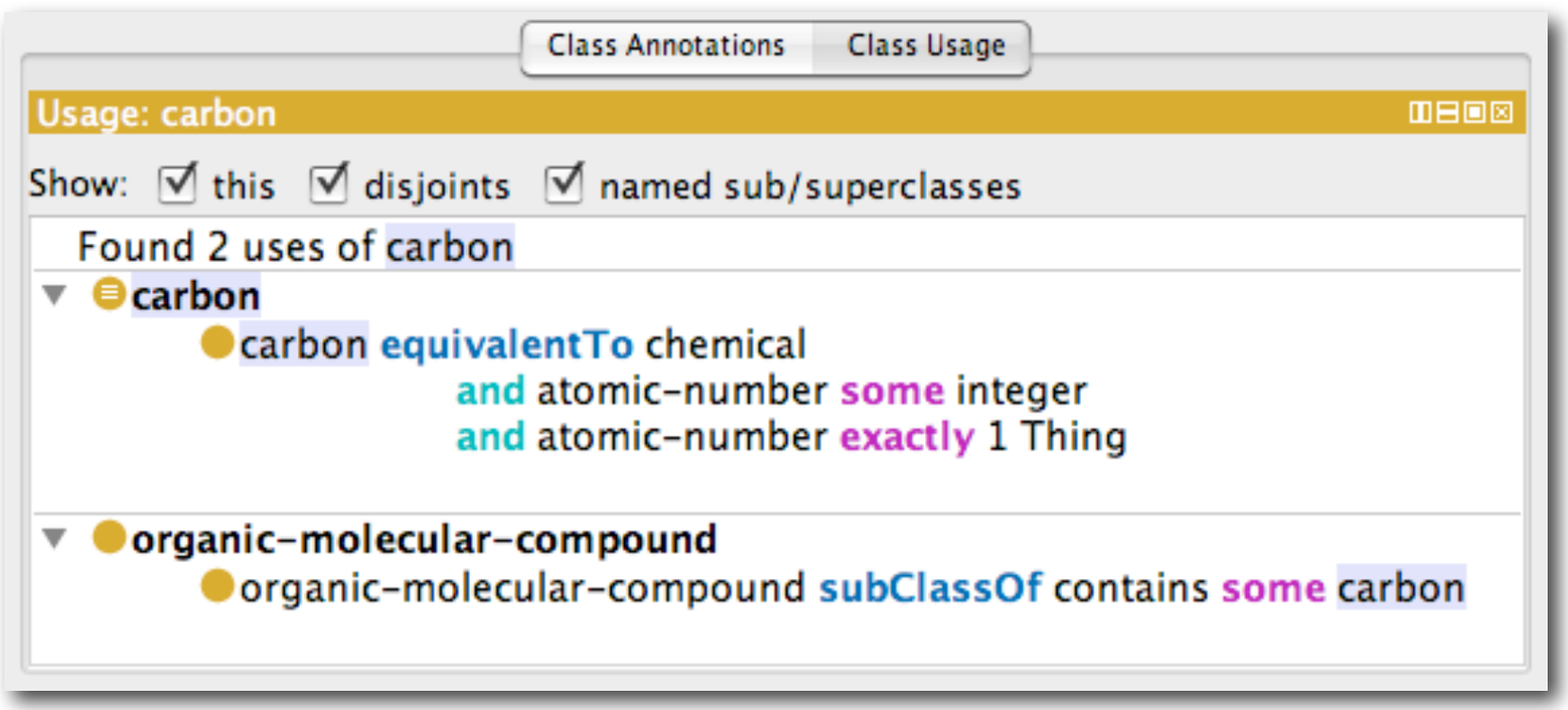
Tedious for big ontologies! ☹

May have comprehension benefits! ☺

Tracing in the old days

Now why is the root *carbon* unsatisfiable?

- Axioms may be spread throughout the ontology.
- Reasons for unsat may be highly non-obvious.



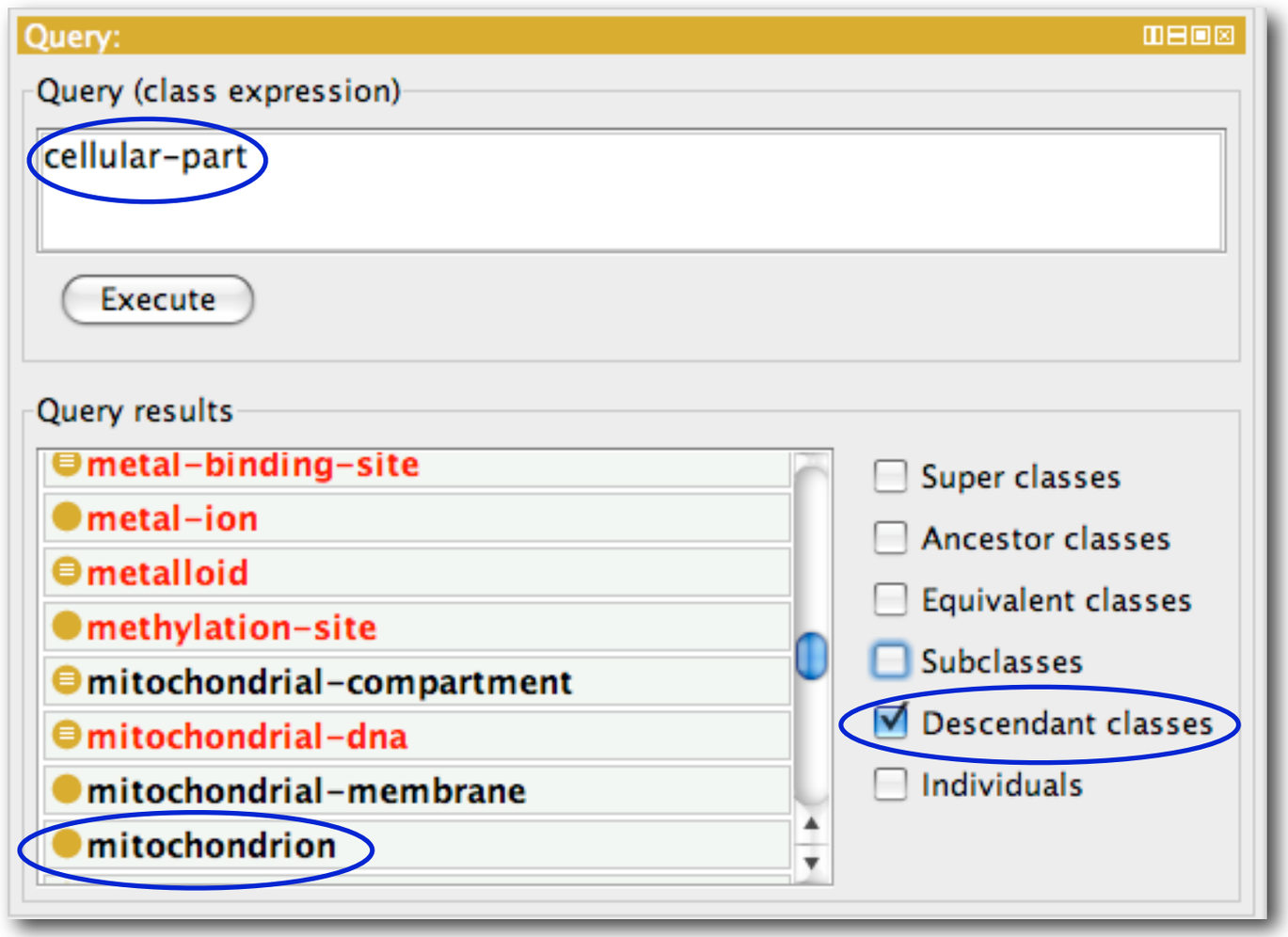
Other entailments

What about inferred subsumptions between two class names?

mitochondrion

$\sqsubseteq$

cellular-part



Tracing difficult and not obvious

Tracing Issues

- Identifies problem classes/terms
  - Not problem axioms
- Not all problem axioms "stick" with terms
  - Tracing can't find these
  - Depends on definition view of tool
  - Binary search of ontology?
- Multiple problem classes
  - Multiple sets of problem axioms

Tracing Issues

- Identifies problem classes/terms
  - Not problem axioms
- Not all problem axioms "stick" with terms
  - Tracing can't find these
  - Depends on definition view of tool
  - Binary search of ontology?
- Multiple problem classes
  - Multiple sets of problem axioms

Navigational approaches insufficient

Explanation

- Reasons for entailments are often difficult to understand. ~> Tool support is required.
- Theoretical basis of explanation has been investigated. New services have been defined.
- Tool support has gone from *None* to *Respectable*.
- It's now fairly easy to get explanations.
  With focus: **Justifications**