

Simplified OWL Ontology Editing

Is WebProtégé Enough?

Matthew Horridge, Tania Tudorache, Jennifer Vendetti
Csongor Nyulas, Mark A. Musen and Natalya F. Noy

Stanford University

Protégé's too complicated!



Main Goals

To develop a **default simple interface** for editing OWL ontologies

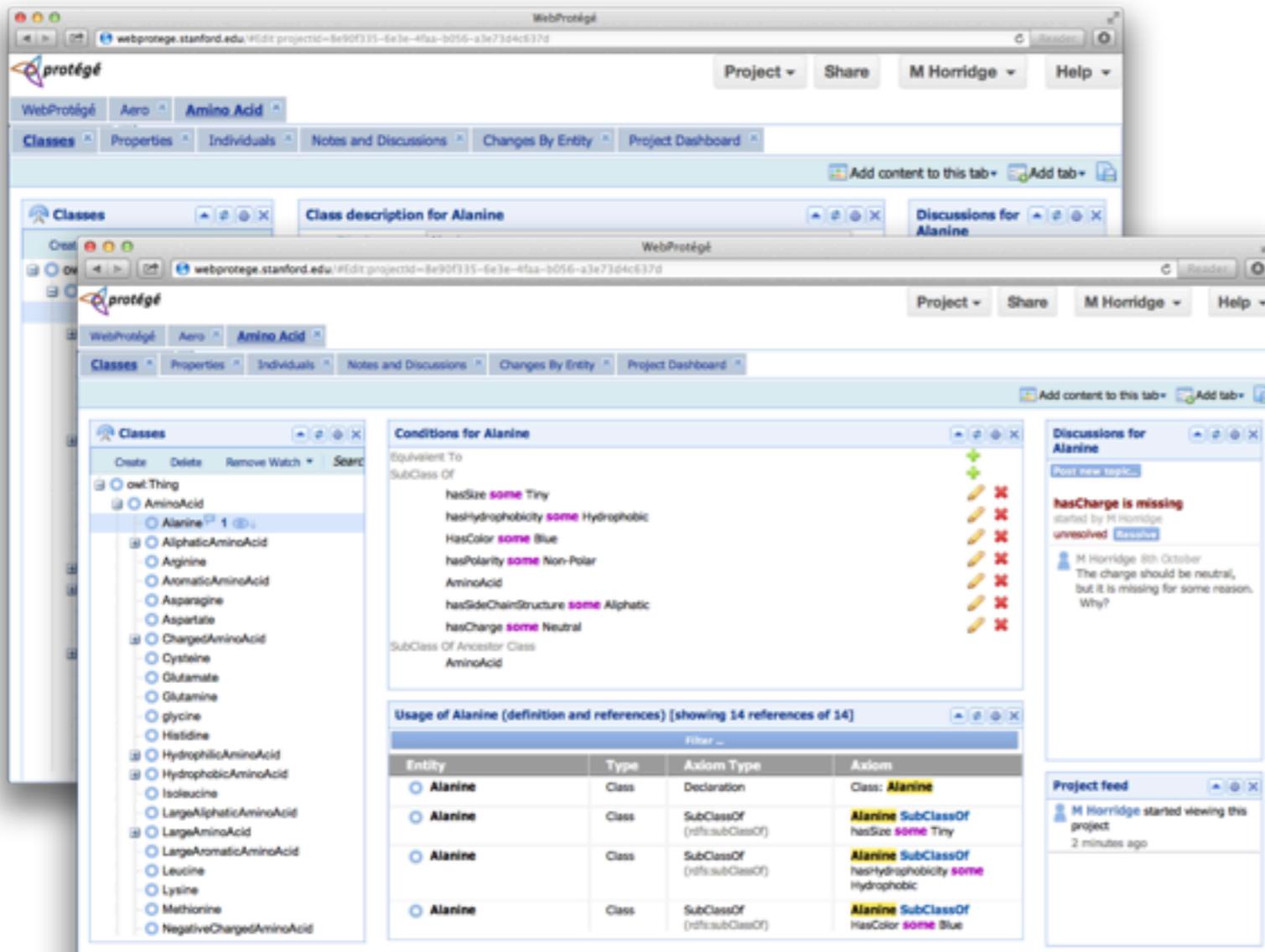
Reduce the intimidation factor for OWL neophytes

Reduce choices that have to be made by OWL neophytes

Make (simple) ontology editing a more pleasant experience

WebProtégé

“Google Docs for Ontologies”



Web-based ontology editor

Configurable user interface

Collaborative editing support

WebProtégé 2.0

May
2013

OWL 2 editing support

Public **ontology** creation & upload

Conjecture

The majority of domain ontologies contain relatively large subsets of axioms that are syntactically simple.

It's possible to design and optimise a simple, but useful, UI for these axioms.

Conjecture

The majority of domain ontologies contain relatively large subsets of axioms that are syntactically simple.

It's possible to design and optimise a simple, but useful, UI for these axioms.

If true,

how can we use this information to design an editor?

Our focus

editing “Class Frames”

The set of axioms that are written down to describe a class

Our focus

editing “Class Frames”

The set of axioms that are written down to describe a class

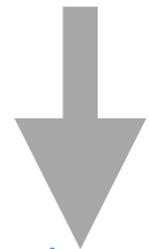
The screenshot shows a software interface for editing a class frame. The title bar reads "Description: GrandFather". The main content area is divided into three sections:

- Equivalent To** (with a plus icon):
 - **Person** (with icons: ?, @, x, o)
 - and (isFatherOf some (Person and (isParentOf some Person)))
- SubClass Of** (with a plus icon):
 - **Person** (with icons: ?, @, x, o)
- SubClass Of (Anonymous Ancestor)**:
 - **hasBirthYear some xsd:integer** (with icons: ?, @, x, o)
 - **hasFather some Man** (with icons: ?, @, x, o)
 - **hasSex some Sex** (with icons: ?, @, x, o)
 - **hasParent max 2 owl:Thing** (with icons: ?, @, x, o)
 - **hasMother some Woman** (with icons: ?, @, x, o)

Empirical Design Methodology

Examine a “seed” corpus

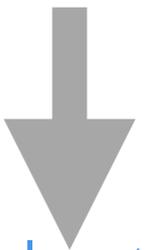
An example of modelling in practice.



dominant types of class axioms and class constructors

Design a “profile” (The WebProtégé Profile).

Design and optimise a UI for this profile



deploy the user interface (deploy WebProtégé)

Evaluate how well the UI covers ontologies uploaded to WebProtégé

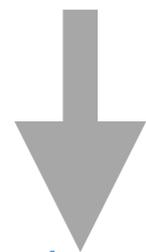
i.e. how useful is the UI in a more general setting?

Quantitative analysis of ontologies + qualitative user study

Empirical Design Methodology

Examine a “seed” corpus

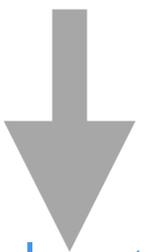
An example of modelling in practice.



dominant types of class axioms and class constructors

Design a “profile” (The WebProtégé Profile).

Design and optimise a UI for this profile

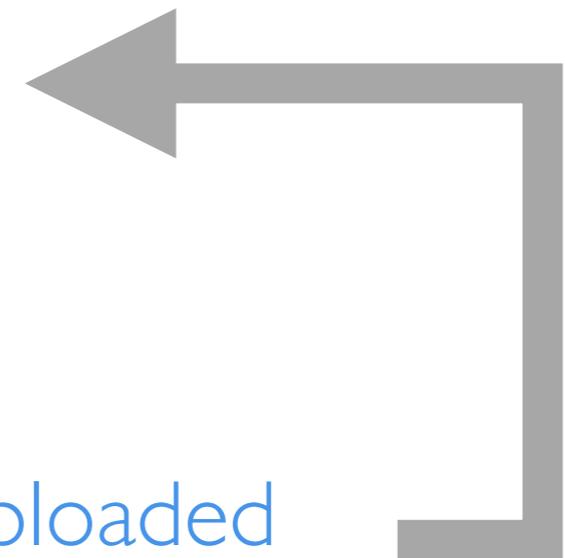


deploy the user interface (deploy WebProtégé)

Evaluate how well the UI covers ontologies uploaded to WebProtégé

i.e. how useful is the UI in a more general setting?

Quantitative analysis of ontologies + qualitative user study



Refine & evolve UI

Our “seed” corpus

The BioPortal Corpus

BioPortal

<http://bioportal.bioontology.org>

The screenshot shows the BioPortal website interface. The browser address bar displays bioportal.bioontology.org/ontologies. The navigation menu includes links for Browse, Search, Mappings, Recommender, Annotator, Resource Index, and Projects. The main content area is titled "Browse" and features a sidebar with filter options (FILTE) and ontology categories (ONTO). A large yellow callout box with a torn edge contains the following text:

300+ biomedical ontologies
More than just class hierarchy/annotations
Wide range of authors
Regularly updated

Below the callout box, a table lists several ontologies with their respective details:

Ascomycete Phenotype Ontology APO	Public	332	0	0	0	10/10/2013	Saccharomyces Genome Database
Basic Formal Ontology BFO	Public	39	0	0	18	07/24/2009	Barry Smith
Beta Cell Genomics Ontology OBI_BCGO	Public	2,450	0	0	1	10/05/2013	Jie Zheng
bilingual Ontology of Alzheimer's Disease OntoAD	Public	624	0	0	0	09/27/2013	

Procedure

① Download **OWL** and **OBO** Ontologies



② Parse each ontology



③ Examine occurrences of **class axioms**:
(SubClassOf, DisjointClasses and EquivalentClasses)



④ Examine occurrences of **class constructors**:
(SomeValuesFrom, AllValuesFrom, Cardinality, Nominals etc. etc)

Procedure

① Download **OWL** and **OBO** Ontologies



② Parse each ontology

261 ontologies



③ Examine occurrences of **class axioms**:

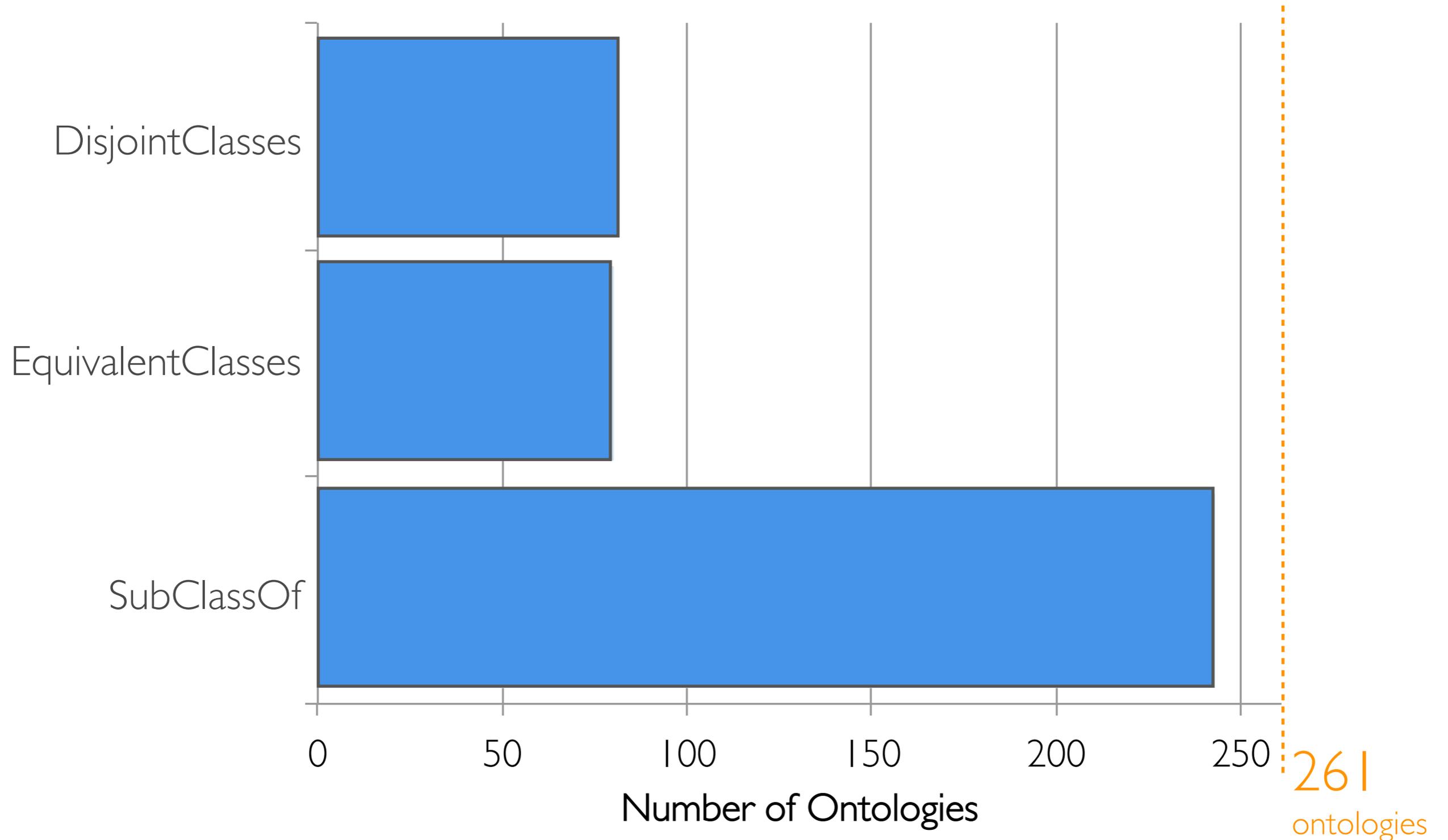
(SubClassOf, DisjointClasses and EquivalentClasses)



④ Examine occurrences of **class constructors**:

(SomeValuesFrom, AllValuesFrom, Cardinality, Nominals etc. etc)

Axiom Type Spread

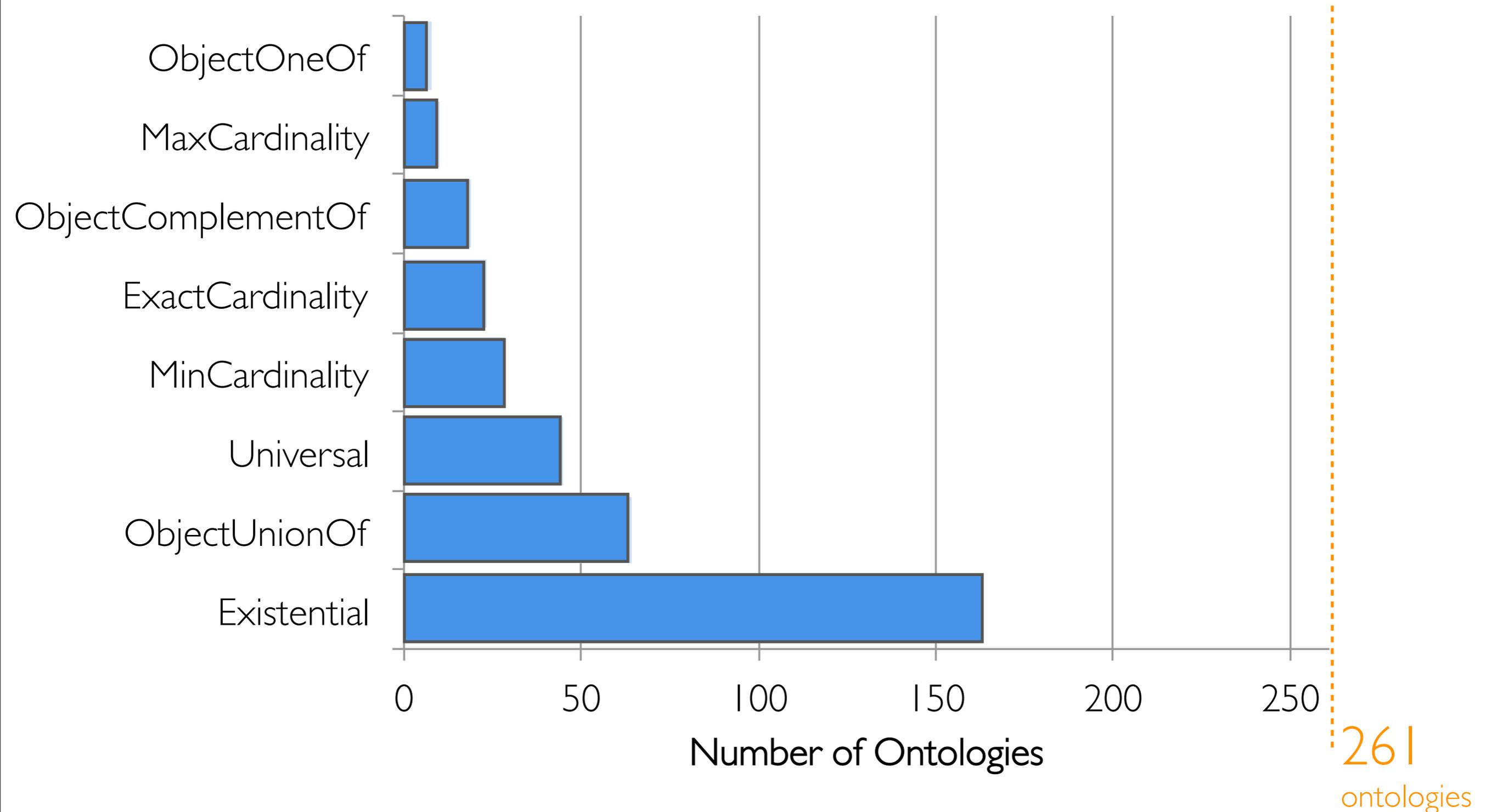


Axiom Types

Occurrences per Ontology

AxiomType	Median	Lower Quartile	Upper Quartile	Maximum
SubClassOf	521	185	2,705	847,755
EquivalentClasses	16	4	61	73,461
DisjointClasses	28	3	158	56,192

Class Expression Spread



Class Expression

Occurrences per Ontology

ClassExpression Type	Median	Lower Quartile	Upper Quartile	Maximum
Existential	157	37	1,461	641,123
Universal	22	4	49	22,371
UnionOf	7	3	20	387

.....▶ focus on

SubClassOf axioms

Class expressions which are existential restrictions

The WebProtégé Profile (WPP)

Definition 2 (WPP). A *WebProtégé Profile class frame* for a class A in the signature of an ontology \mathcal{O} is the subset-maximal set of axioms $\mathcal{S} \subseteq \mathcal{O}$ such that each axiom in \mathcal{S} conforms to the following grammar, where non-terminals are shown in bold, terminals are shown in a regular font-weight surrounded by single quotes, choices are indicated with a bar, zero or more items are shown in curly brackets. The non-terminals **Class**, **ObjectProperty**, **DataProperty**, **AnnotationProperty**, **NamedIndividual**, **Datatype**, **Literal** and **IRI**, are defined as they appear in the OWL 2 Structural Specification.

ClassFrame := { **ClassFrameAxiom** }

ClassFrameAxiom := 'SubClassOf' (' A **ClassExpression** ') |
'AnnotationAssertion' (' **AnnotationProperty** A **AnnoValue** ')

ClassExpression := **Class** |
'ObjectIntersectionOf' (' **ClassExpression** **ClassExpression** { **ClassExpression** } ') |
'ObjectSomeValuesFrom' (' **ObjectProperty**, **Class** ') |
'ObjectSomeValuesFrom' (' **ObjectProperty**, { ' **NamedIndividual** } ') |
'ObjectHasValue' (' **ObjectProperty**, **NamedIndividual** ') |
'DataSomeValuesFrom' (' **DataProperty**, **Datatype** ') |
'DataSomeValuesFrom' (' **DataProperty**, { ' **Literal** } ') |
'DataHasValue' (' **DataProperty**, **Literal** ') |
'ObjectMinCardinality' (' '1' **ObjectProperty**, **Class** ') |
'DataMinCardinality' (' '1' **DataProperty**, **Class** ')

AnnoValue := **Literal** | **IRI**

The WebProtégé Profile (WPP)

Definition 2 (WPP). A WebProtégé Profile class frame for a class A in the signature of an ontology \mathcal{O} is the subset-maximal set of axioms $\mathcal{S} \subseteq \mathcal{O}$ such that each axiom in \mathcal{S} conforms to the following grammar, where non-terminals are shown in bold, terminals are shown in a regular font-weight surrounded by single quotes, choices are indicated with a bar, zero or more items are shown in curly brackets. The non-terminals **Class**, **ObjectProperty**, **DataProperty**, **AnnotationProperty**, **NamedIndividual**, **Datatype**, **Literal** and **IRI**, are defined as they appear in the OWL 2 Structural Specification.

```
ClassFrame := { ClassFrame }  
ClassFrameAxiom := 'SubClassOf' (' A ClassExpression ') |  
                    'AnnotationAssertion' (' AnnotationProperty A AnnoValue ')  
ClassExpression := Class |  
                    'ObjectIntersectionOf' (' ClassExpression ClassExpression { ClassExpression } ') |  
                    'ObjectSomeValuesFrom' (' ' ObjectProperty, Class ') |  
                    'ObjectSomeValuesFrom' (' ' ObjectProperty, { ' NamedIndividual } ') |  
                    'ObjectHasValue' (' ' ObjectProperty, NamedIndividual ') |  
                    'DataSomeValuesFrom' (' ' DataProperty, Datatype ') |  
                    'DataSomeValuesFrom' (' ' DataProperty, { ' Literal } ') |  
                    'DataHasValue' (' ' DataProperty, Literal ') |  
                    'ObjectMinCardinality' (' '1' ObjectProperty, Class ') |  
                    'DataMinCardinality' (' '1' DataProperty, Class )  
AnnoValue := Literal | IRI
```

The WebProtégé Profile (WPP)

Definition 2 (WPP). A WebProtégé Profile class frame for a class A in the signature of an ontology \mathcal{O} is the subset-maximal set of axioms $\mathcal{S} \subseteq \mathcal{O}$ such that each axiom in \mathcal{S} conforms to the following grammar, where non-terminals are shown in bold, terminals are shown in a regular font-weight surrounded by single quotes, choices are indicated with a bar, zero or more items are shown in curly brackets. The non-terminals **Class**, **ObjectProperty**, **DataProperty**, **AnnotationProperty**, **NamedIndividual**, **Datatype**, **Literal** and **IRI**, are defined as they appear in the OWL 2 Structural Specification.

```
ClassFrame := { ClassFrame }  
ClassFrameAxiom := 'SubClassOf' (' A ClassExpression ') |  
                    'AnnotationAssertion' (' AnnotationProperty A AnnoValue ')  
ClassExpression := Class |  
                    'ObjectIntersectionOf' (' ClassExpression ClassExpression { ClassExpression } ') |  
                    'ObjectSomeValuesFrom' (' ObjectProperty, Class ') |  
                    'ObjectSomeValuesFrom' (' ObjectProperty, { ' NamedIndividual ' } ') |  
                    'ObjectHasValue' (' ObjectProperty, NamedIndividual ') |  
                    'DataSomeValuesFrom' (' DataProperty, Datatype ') |  
                    'DataSomeValuesFrom' (' DataProperty, { ' Literal ' } ') |  
                    'DataHasValue' (' DataProperty, Literal ') |  
                    'ObjectMinCardinality' (' '1' ObjectProperty, Class ') |  
                    'DataMinCardinality' (' '1' DataProperty, Class )  
AnnoValue := Literal | IRI
```

Evaluation Coverage & Usability

Reminder

☀ Create Project ...

🏠 Upload Project ...

Two modes of operation:
Create ontologies from scratch
or
Upload existing ontologies

The WebProtégé Corpus

230 uploaded pre-existing ontologies

Most uploaded between 25th April 2013 - 6th May 2013

519 ontologies in total

230 uploaded pre-existing ontologies

- 1 Parse each ontology
- 2 Count class frames in WPP
- 3 Count class frames in WPP-Dis
(WPP + DisjointClasses axioms)
- 4 Count class frames in WPP-Dis-Eq
(WPP + DisjointClasses and EquivalentClasses axioms)

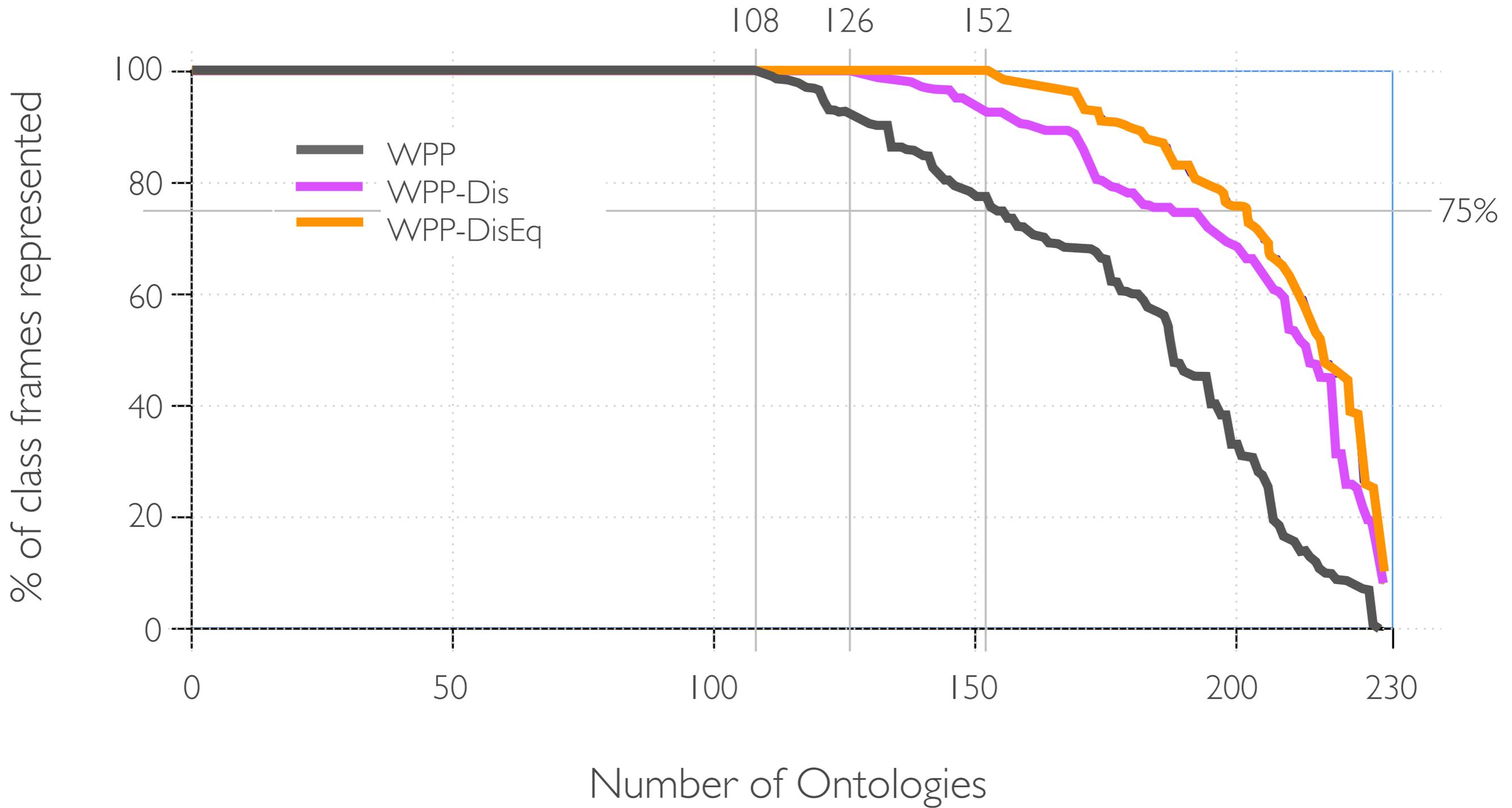
230 pre-existing ontologies uploaded

- 1 Parse each ontology
- 2 Count class frames in WPP
- 3 Count class frames in WPP-Dis
(WPP + DisjointClasses axioms)
- 4 Count class frames in WPP-Dis-Eq
(WPP + DisjointClasses and EquivalentClasses axioms)

230 pre-existing ontologies uploaded

- 1 Parse each ontology
- 2 Count class frames in WPP
- 3 Count class frames in WPP-Dis
(WPP + DisjointClasses axioms)
- 4 Count class frames in WPP-Dis-Eq
(WPP + DisjointClasses and EquivalentClasses axioms)

Coverage over WP Corpus



The UI

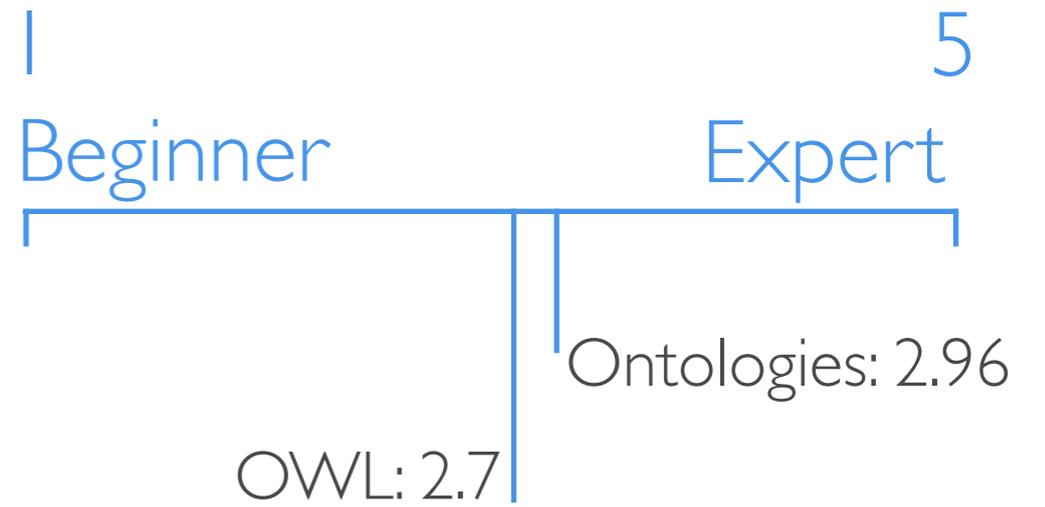
Properties

<input type="checkbox"/> hasFlightControlSystem	<input type="radio"/> Fly By Wire System	<input type="checkbox"/>
<input type="checkbox"/> hasManufacturer	<input type="radio"/> AirbusIndustrie	<input type="checkbox"/>
<input type="checkbox"/> hasMaxPassengerCapacity	# 180	<input type="checkbox"/>
hasFlap	DoubleSlottedFlap	<input type="checkbox"/>
<div style="background-color: yellow; padding: 5px;">! hasFlap is a new property name. To continue, enter a value for the property (class name, number etc.) and press the TAB key</div>	<div style="border: 1px solid gray; padding: 5px;">New Class named DoubleSlottedFlap New NamedIndividual named DoubleSlottedFlap New Datatype named DoubleSlottedFlap</div>	
Enter property name	Enter value	<input type="checkbox"/>

User Study

55 Participants: WebProtégé users

Recruited via direct invitation and mailing list

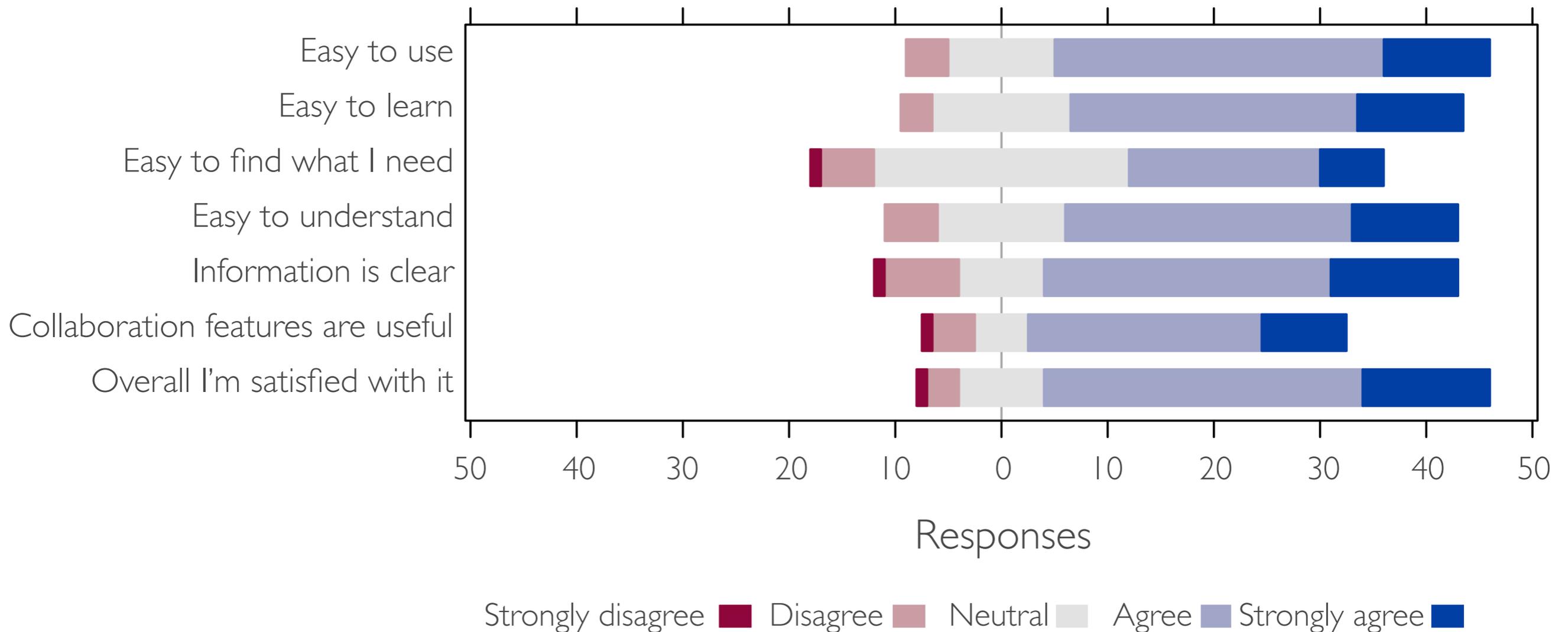
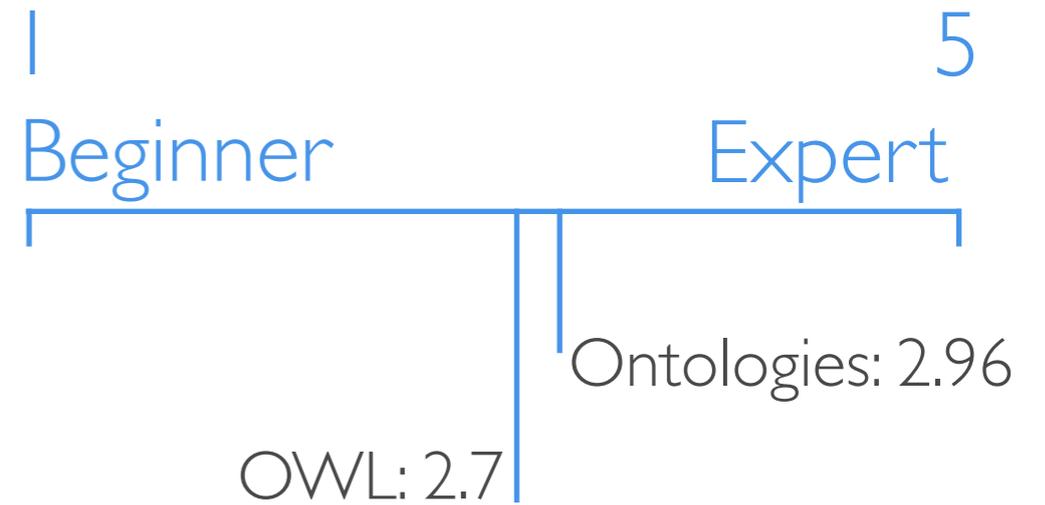


User Study

55 Participants: WebProtégé users

Recruited via direct invitation and mailing list

Standard Nielsen usability questions



What do you miss?

| User: "The ability to create anonymous classes"

| User: "The ability to create logical expressions"

Summary

An empirically driven UI development methodology for designing a simplified user interface for editing OWL ontologies

WebProtégé Profile (WPP) - a simple syntactic fragment of OWL 2

A simple UI to support this profile - the default UI in WebProtégé

The profile/UI offers good coverage for naturally occurring ontologies, and we have received positive feedback about the UI from users

We found the empirical design methodology useful

Classes

Create Delete Watch Branch ▾

- owl:Thing
 - AminoAcid
 - AliphaticAminoAcid
 - Alanine 3
 - Asparagine
 - Cysteine
 - Glycine
 - LargeAliphaticAminoAcid
 - NegativeChargedAminoAcid
 - Proline
 - TinyPolarAminoAcid
 - Valine
 - AromaticAminoAcid
 - LargeAliphaticAminoAcid
 - NegativeChargedAminoAcid
 - PositiveChargedAminoAcid
 - TinyPolarAminoAcid
 - RefiningFeature
 - SpecificAminoAcid

Class description for Alanine

Display name Alanine

IRI <http://www.co-ode.org/ontologies/amino-acid/2006/05/18/amino-acid#Alanine>

Annotations

rdfs:label	Alanine	✕
hasPubChemNumber	# 5950	✕
shortName	ala	✕
Enter property name	Enter value	

Properties

hasHydrophobicity	Hydrophobic	✕
hasPolarity	Non-Polar	✕
hasSideChainStructure	Aliphatic	✕
hasSize	Tiny	✕
Enter property name	Enter value	

Discussions for Alanine

Post new topic...

Add a seeAlso link

started by M Horridge

resolved Re-open

M Horridge 13th October
Add a see also link which points to the wikipedia page:
<http://en.wikipedia.org/wiki/Alanine>

Missing charge description

started by M Horridge

unresolved Resolve

M Horridge 13th October
This is missing an axiom stating that it has a charge.

M Horridge 13th October
By the way, it's **neutral**

This work

Goal: To develop a **simple interface** for editing OWL ontologies

Empirically driven UI design methodology

Base the design on the use of language constructs in an existing ontology corpus

This work

Goal: To develop a **simple interface** for editing OWL ontologies

Empirically driven UI design methodology

Base the design on the use of language constructs in an existing ontology corpus

... and, try to minimise choices that need to be made by users