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
OWL 2 editing support
Public ontology creation & upload

May 2013
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

Description: GrandFather

Equivalent To

- Person
  - and (isFatherOf some (Person and (isParentOf some Person)))

SubClass Of

- Person

SubClass Of (Anonymous Ancestor)

- hasBirthYear some xsd:integer
- hasFather some Man
- hasSex some Sex
- hasParent max 2 owl:Thing
- hasMother some Woman
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

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

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

Number of Ontologies

DisjointClasses
EquivalenceClasses
SubClassOf

261 ontologies
<table>
<thead>
<tr>
<th>Axiom Type</th>
<th>Median</th>
<th>Lower Quartile</th>
<th>Upper Quartile</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubClassOf</td>
<td>521</td>
<td>185</td>
<td>2,705</td>
<td>847,755</td>
</tr>
<tr>
<td>EquivalentClasses</td>
<td>16</td>
<td>4</td>
<td>61</td>
<td>73,461</td>
</tr>
<tr>
<td>DisjointClasses</td>
<td>28</td>
<td>3</td>
<td>158</td>
<td>56,192</td>
</tr>
</tbody>
</table>
Class Expression Spread

Number of Ontologies

ObjectOneOf
MaxCardinality
ObjectComplementOf
ExactCardinality
MinCardinality
Universal
ObjectUnionOf
Existential

Number of Ontologies

261 ontologies
<table>
<thead>
<tr>
<th>ClassExpression Type</th>
<th>Median</th>
<th>Lower Quartile</th>
<th>Upper Quartile</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existential</td>
<td>157</td>
<td>37</td>
<td>1,461</td>
<td>641,123</td>
</tr>
<tr>
<td>Universal</td>
<td>22</td>
<td>4</td>
<td>49</td>
<td>22,371</td>
</tr>
<tr>
<td>UnionOf</td>
<td>7</td>
<td>3</td>
<td>20</td>
<td>387</td>
</tr>
</tbody>
</table>
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 \( O \) is the subset-maximal set of axioms \( S \subseteq O \) such that each axiom in \( S \) 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 \( \text{Class}, \text{ObjectProperty}, \text{DataProperty}, \text{AnnotationProperty}, \text{NamedIndividual}, \text{Datatype}, \text{Literal} \) and \( \text{IRI} \), are defined as they appear in the OWL 2 Structural Specification.

\[
\begin{align*}
\text{ClassFrame} & := \{ \text{ClassFrameAxiom} \} \\
\text{ClassFrameAxiom} & := \text{`SubClassOf`} \ '(\text{"A ClassExpression"})' | \\
& \quad \text{`AnnotationAssertion`} \ '(\text{"AnnotationProperty A AnnoValue"})' \\
\text{ClassExpression} & := \text{Class} | \\
& \quad \text{`ObjectIntersectionOf`} \ '(\text{"ClassExpression ClassExpression \{ClassExpression\}"})' | \\
& \quad \text{`ObjectSomeValuesFrom`} \ '(\text{"ObjectProperty, Class"})' | \\
& \quad \text{`ObjectSomeValuesFrom`} \ '(\text{"ObjectProperty, \{\text{NamedIndividual}\}"})' | \\
& \quad \text{`ObjectHasValue`} \ '(\text{"ObjectProperty, NamedIndividual"})' | \\
& \quad \text{`DataSomeValuesFrom`} \ '(\text{"DataProperty, Datatype"})' | \\
& \quad \text{`DataSomeValuesFrom`} \ '(\text{"DataProperty, \{\text{Literal}\}"})' | \\
& \quad \text{`DataHasValue`} \ '(\text{"DataProperty, Literal"})' | \\
& \quad \text{`ObjectMinCardinality`} \ '(\text{"1\ ObjectProperty, Class"})' | \\
& \quad \text{`DataMinCardinality`} \ '(\text{"1\ DataProperty, Class"})' \\
\text{AnnoValue} & := \text{Literal} | \text{IRI}
\end{align*}
\]
The WebProtégé Profile (WPP)

Definition 2 (WPP). A WebProtégé Profile class frame for a class \( A \) in the signature of an ontology \( O \) is the subset-maximal set of axioms \( S \subseteq O \) such that each axiom in \( S \) 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 \( \text{Class} \), \( \text{ObjectProperty} \), \( \text{DataProperty} \), \( \text{AnnotationProperty} \), \( \text{NamedIndividual} \), \( \text{Datatype} \), \( \text{Literal} \) and \( \text{IRI} \), are defined as they appear in the OWL 2 Structural Specification.

\[
\begin{align*}
\text{ClassFrame} & := \{ \text{ClassFrameAxiom} \} \\
\text{ClassFrameAxiom} & := \text{SubClassOf} \ ('(' \text{A ClassExpression} ')') \ | \\
& \quad \text{AnnotationAssertion} \ ('(' \text{AnnotationProperty} \text{A AnnoValue} ')') \\
\text{ClassExpression} & := \text{Class} \ | \\
& \quad \text{ObjectIntersectionOf} \ ('(' \text{ClassExpression} \text{ClassExpression} \{ \text{ClassExpression} \} ')') \ | \\
& \quad \text{ObjectSomeValuesFrom} \ ('(' \text{ObjectProperty, Class} ')') \ | \\
& \quad \text{ObjectSomeValuesFrom} \ ('(' \text{ObjectProperty, \{\text{NamedIndividual}\} ')') \ | \\
& \quad \text{ObjectHasValue} \ ('(' \text{ObjectProperty, NamedIndividual} ')') \ | \\
& \quad \text{DataSomeValuesFrom} \ ('(' \text{DataProperty, Datatype} ')') \ | \\
& \quad \text{DataSomeValuesFrom} \ ('(' \text{DataProperty, \{\text{Literal}\} ')') \ | \\
& \quad \text{DataHasValue} \ ('(' \text{DataProperty, Literal} ')') \ | \\
& \quad \text{ObjectMinCardinality} \ ('(' \text{1 ObjectProperty, Class} ')') \ | \\
& \quad \text{DataMinCardinality} \ ('(' \text{1 DataProperty, Class} ')') \\
\text{AnnoValue} & := \text{Literal} \ | \ \text{IRI}
\end{align*}
\]
The WebProtégé Profile (WPP)

Definition 2 (WPP). A WebProtégé Profile class frame for a class $A$ in the signature of an ontology $O$ is the subset-maximal set of axioms $S \subseteq O$ such that each axiom in $S$ 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
```
Evaluation
Coverage & Usability
Reminder

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
   \[(\text{WPP} + \text{DisjointClasses axioms})\]
4. Count class frames in WPP-Dis-Eq
   \[(\text{WPP} + \text{DisjointClasses and EquivalentClasses axioms})\]
Coverage by Ontology

- Class-frames inside the WebProtégé Profile
- Class-frames inside the WebProtégé Profile augmented with DisjointClasses and EquivalentClasses axioms
- Class-frames inside the WebProtégé Profile augmented with DisjointClasses axioms
- Class-frames inside the WebProtégé Profile
Coverage over WP Corpus

% of class frames represented vs. Number of Ontologies

- WPP
- WPP-Dis
- WPP-DisEq

Coverage reaches approximately 75% at around 126 ontologies.
The UI

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasFlightControlSystem</td>
<td>Fly By Wire System</td>
</tr>
<tr>
<td>hasManufacturer</td>
<td>AirbusIndustrie</td>
</tr>
<tr>
<td>hasMaxPassengerCapacity</td>
<td># 180</td>
</tr>
<tr>
<td>hasFlap</td>
<td>DoubleSlottedFlap</td>
</tr>
</tbody>
</table>

**hasFlap** is a new property name.
To continue, enter a value for the property (class name, number etc.) and press the TAB key.

New Class named DoubleSlottedFlap
New NamedIndividual named DoubleSlottedFlap
New Datatype named DoubleSlottedFlap

Enter property name
Enter value
User Study

55 Participants: WebProtégé users
Recruited via direct invitation and mailing list

Beginner

Expert

Ontologies: 2.96
OWL: 2.7
User Study

55 Participants: WebProtégé users
Recruited via direct invitation and mailing list

Standard Nielsen usability questions

- Easy to use
- Easy to learn
- Easy to find what I need
- Easy to understand
- Information is clear
- Collaboration features are useful
- Overall I'm satisfied with it

Responses

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>40</td>
</tr>
</tbody>
</table>

Beginner: 55
Expert: 0
Ontologies: 2.96
OWL: 2.7
What do you miss?

1 User: “The ability to create anonymous classes”

1 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
### Class description for Alanine

<table>
<thead>
<tr>
<th>Display name</th>
<th>Alanine</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRI</td>
<td><a href="http://www.co-ode.org/ontologies/amino-acid/2006/05/18/amino">http://www.co-ode.org/ontologies/amino-acid/2006/05/18/amino</a></td>
</tr>
</tbody>
</table>

#### Annotations
- `rdfs:label`: Alanine
- `hasPubChemNumber`: # 5950
- `shortName`: ala

#### Properties
- `hasHydrophobicity`: Hydrophobic
- `hasPolarity`: Non-Polar
- `hasSideChainStructure`: Aliphatic
- `hasSize`: Tiny

### Discussions for Alanine

#### Add a seeAlso link
- Started by M Horridge
- Resolved
- Re-open

- **Add a seeAlso link**
  - Add a see also link which points to the wikipedia page: [http://en.wikipedia.org/wiki/Alanine](http://en.wikipedia.org/wiki/Alanine)

#### Missing charge description
- Started by M Horridge
- Unresolved
- Resolve

- **Missing charge description**
  - This is missing an axiom stating it has a charge.

- **Missing charge description**
  - 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