OntoTrack
A New Ontology Authoring Approach

Thorsten Liebig and Olaf Noppens

Dept. of Artificial Intelligence
University of Ulm, D-89069 Ulm
{liebig|noppens}@informatik.uni-ulm.de

1 Motivation

List-based expand and contract tree-like interfaces for ontologies inherently have substantial drawbacks concerning search and navigation speed as well as user orientation and editing efficiency. As a consequence many ontology editors use two functionally disjunct interfaces for either editing or browsing ontologies.

OntoTrack implements a novel approach using one integrated view for browsing and editing optimized for navigation and manipulation of large OWL Lite ontologies. More concrete, OntoTrack

- Provides a sophisticated graph-based ontology layout with animated expansion and de-expansion of class descendants, zooming, and paning.
- Uses elaborated layout techniques like click-able miniature branches or selective detail views.
- Allows for editing features like mouse-over anchor buttons, graphical selections or restriction editing without switching into a special editing layout.
- Synchronizes every single editing step with an external reasoner in order to provide instant feedback about relevant modeling consequences.
- Implements instant search highlighting or an overlay representation of classes and properties.

Since it is a pure Java application it runs on almost all Java-enabled platforms (tested on Windows, Linux, Solaris, and MacOS X yet). The system can be downloaded at `http://www.informatik.uni-ulm.de/ki/ontotrack` under a non-commercial use license.

2 OntoTrack System Description

The following provides a short description of OntoTracks main features. For a more detailed report see [1]. A graphical review is given in Figure 2.

2.1 Layouting Ontologies According to their Nature

The primary structuring element of ontologies is the subsumption relationship between classes. Consequently, OntoTrack layouts an ontology as a directed acyclic graph (DAG) with classes as nodes and the subsumption relationship as edges either in top-down, left-right, bottom-up, or right-left orientation. The drawn hierarchy may vary from the explicitly given subclass statements of the ontology source document. This is because OntoTrack uses an external OWL reasoner called RACER in order to make all implicit subsumption relationships explicity visible to the user.

The direct descendants of a class can be expanded as well as de-expanded in an animated fashion on left mouse button click. OntoTrack's ontology layout is driven by the expansion direction of the user. The layout strategy aims at "clustering" the descendants of a selected class (comparable to an ordinary tree representation). This means, that the same classes of an ontology with multiple inheritance may be layouted different – depending on the users expansion order.

The screen shot in Figure 1 shows a fraction of the OpenCyc ontology (version 0.7) in top-down orientation. Note, that the subsumption path of the currently selected class (PartIntangibleIndiv) up to the OWL root class (Thing) is outlined by a darker node background and thicker edges in order to provide an optimal overview concerning depth and branching.

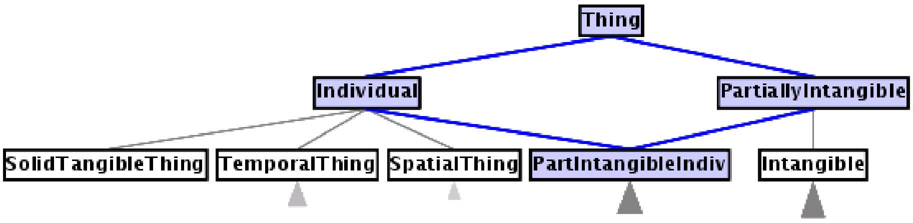

Fig. 1. Some top concepts of the OpenCyc ontology in top-down orientation.

OntoTrack provides a likewise browsing and editing functionality for properties. In addition, one hierarchy can be rendered as a read-only transparent layer on top of the other.

2.2 Maximizing Information while Minimizing Information Burden

Not expanded sub-branches of classes are symbolized by triangles of varying length, width, and shading, approximating the depth, branching and number of subclasses, or as miniature sub-branches (depending on the number of descendants). The icons of miniature sub-branches are expandable on mouse click and show their class names via mouse-over tool-tip.

OntoTrack always shows all (even implicit) direct subsumption relationships between classes. However, when dealing with classes having multiple ancestors it might be the case that some ancestors are not visible (because they are in not expanded branches). Those ancestors are drawn as click-able thumbnail classes in order to show the semantically correct hierarchy while minimizing the number of expanded classes at the same time. OntoTrack will not expand them by default because of the assumption, that the user is much more interested in the descendants of the actual class (he/she actually asked for) instead of their ancestors.

2.3 Manipulating via Mouse Action

A new sub- or superclass can easily be added by using the anchor mode of OntoTrack. Here, click-able triangle buttons appear when moving with the mouse pointer over class nodes. These anchor buttons allow for specifying an additional superclass or new subclass and are sensible with respect to the layout orientation. Class identifiers are easily be changed on double mouse click. An existing superclass relationship can be changed via mouse drag-and-drop or deleted by selection and "Del"-key usage.

2.4 Editing without Syntactic Hurdles

In addition, OntoTrack offers further editing functionality while in its "detailed view" mode. The detailed view mode is (de-)activated for each class separately using the mouse-wheel up- resp. down-wards while being over the class with the mouse pointer. When activated, OntoTrack uses an UML-style class diagram showing the list of defined property restrictions for this class in abstract Description Logic (DL) syntax. In general, a restriction can be deleted or added to a class definition simply by clicking on corresponding buttons within the UML class box. Quantifiers, cardinalities, properties as well as classes within those restrictions are chosen with help of selection lists.

2.5 Instant Feedback about Modeling Consequences

All changes after each editing step (e.g. list selection, subclass statement) are send to an OWL reasoner. This reasoner will then make all modeling consequences explicitly available. OntoTrack will hand over relevant consequences to the user by providing appropriate graphical feedback. Besides the already mentioned subsumption determination OntoTrack highlights all those classes which are unsatisfiable or equivalent according to the underlying semantics.

2.6 Search-Down without Pop-Up's

OntoTrack also implements a string matching search with graphical highlighting. Each search string change directly results in an updated highlighting of matching parts of the ontology (expanded or not).

References

1. Liebig, T., Noppens, O.: OntoTrack: Combining Browsing and Editing with Reasoning and Explaining for OWL Lite Ontologies. In: Proc. of the 3rd International Semantic Web Conference (ISWC), Hiroshima, Japan (2004) to appear.

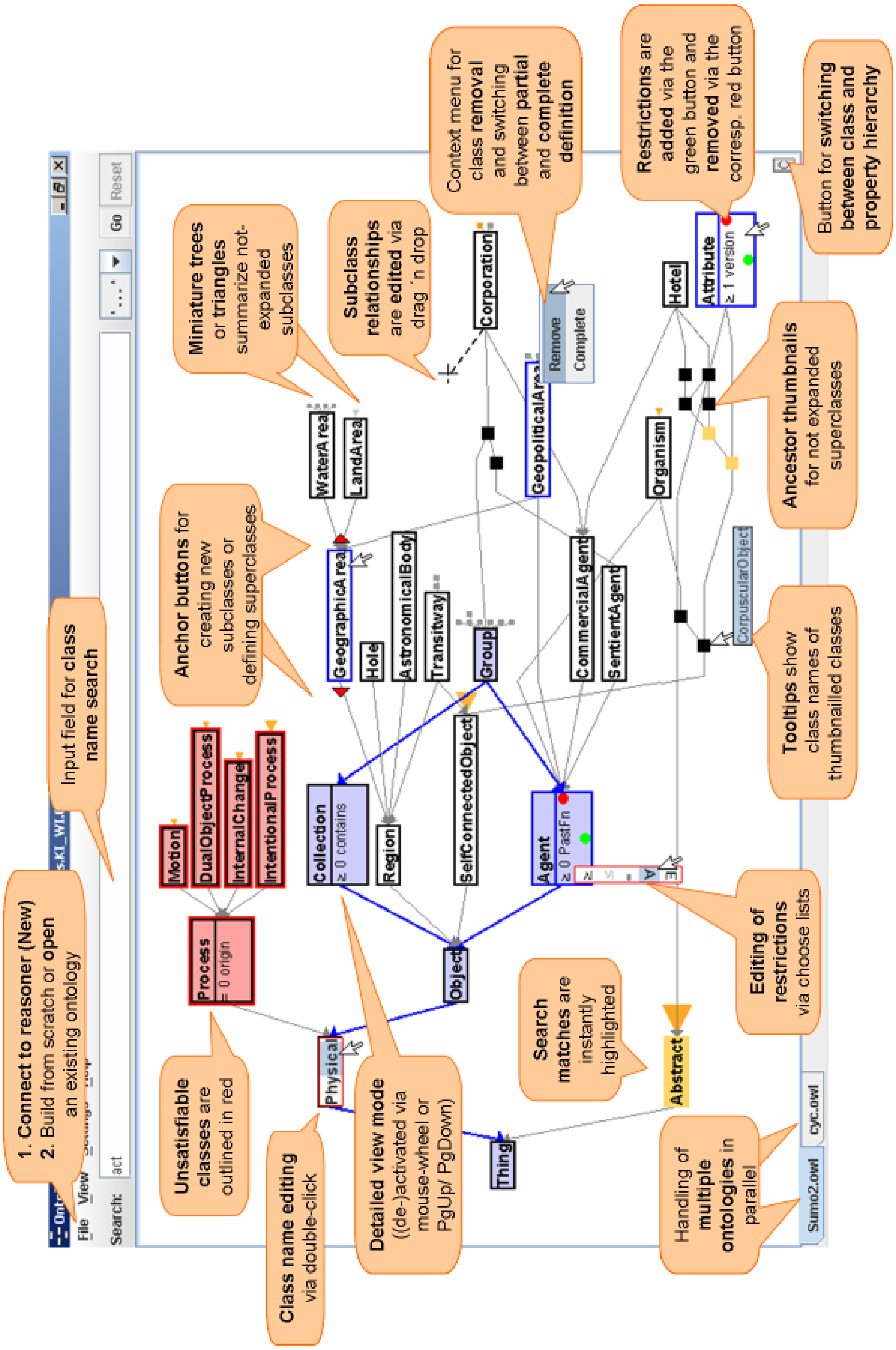
1. Connect to reasoner (New)
2. Build from scratch or open an existing ontology
Input field for class name search
Anchor buttons for creating new subclasses or defining superclasses
Miniature trees or triangles summarize not-expanded subclasses
Subclass relationships are edited via drag 'n drop
Context menu for class removal and switching between partial and complete definition
Restrictions are added via the green button and removed via the corresp. red button
Button for switching between class and property hierarchy
Ancestor thumbnails for not expanded superclasses
Tooltips show class names of thumbnailed classes
Editing of restrictions via choose lists
Search matches are instantly highlighted
Handling of multiple ontologies in parallel
Detailed view mode ((de-)activated via mouse-wheel or PgUp/ PgDown)
Class name editing via double-click
Unsatisfiable classes are outlined in red
File View
Search: act
Go Reset
Remove
Complete
Sumo2.owl
cyc.owl
Thing
Physical
Abstract
Object
Process
≥ 0 origin
Motion
DualObjectProcess
InternalChange
IntentionalProcess
Collection
≥ 0 contains
Region
SelfConnectedObject
Agent
≥ 0 PastFn
GeographicArea
WaterArea
LandArea
Hole
AstronomicalBody
Transitway
Group
CommercialAgent
SentientAgent
GeopoliticalArea
Corporation
Organism
Hotel
Attribute
≥ 1 version
CorpuscularObject

Fig. 2. Overview over the main features of OntoTrack.