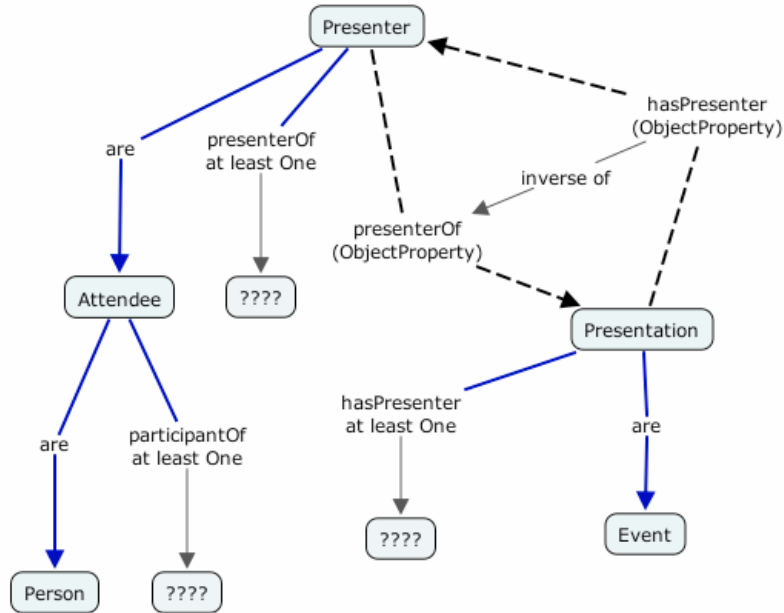


# Concept-map Ontology Environment

*Concept mapping* is a tested, intuitive, low-entry-cost technique for knowledge capture and composition.

OWL is the new Semantic Web standard for formal knowledge representation and communication between systems.



**COE brings these worlds together.**

COE uses concept maps to display, edit and compose OWL, in an integrated GUI combining Cmap display with concept search and cluster analysis.

## The Semantic Web as a distributed information environment

The Semantic Web vision is of a network of machine-useable content all openly published and communicable on a network (not *necessarily* the WWWeb.)

A key part of this vision is that authors of new content can re-use appropriate concepts which already occur in other content; this reinforces the meaning of those concepts by relating them to new content. Such re-use provides a ***distributed syndication*** model that avoids classical problems of concept mapping between rival formalizations.

Examples of this include the Dublin Core vocabulary and the FOAF project, both of which obtain their meanings primarily from their user community.

## The Semantic Web as a distributed information environment

The Semantic Web vision is of a network of **machine-useable content ...**

Q: What can machines use it to do?

- A:
- draw conclusions;
  - recognize patterns in data;
  - convert between data formats;
  - relate concept definitions to data;
  - do smart retrieval of relevant texts;
  - and so on.

**In brief: *help you access what is of interest to you.***

If you know about concept mapping.....

COE is built on top of IHMC Concept Map Tools tool suite, so (almost) anything you can do in normal concept mapping you can also do in COE: adding nodes, moving things around, dragging and clicking, navigating, etc..

If you know about concept mapping.....

COE is built on top of IHMC Concept Map Tools tool suite, so (almost) anything you can do in normal concept mapping you can also do in COE: adding nodes, moving things around, dragging and clicking, navigating, etc..

**BUT** in order to understand the imported OWL ontologies, and produce Cmaps which will be output as OWL, you will need to learn, and use, some new conventions.

The COE GUI provides a number of new tricks to make your task easier.

And COE is also a Web-oriented concept search and clustering tool.

## If you know about OWL and the Semantic Web.....

COE allows you to view, edit and compose OWL ontologies without needing to know anything about, or ever look at, XML or RDF. Just *looking* at OWL ontologies with COE is often very illuminating. The concept map GUI can be learned by SWeb newbies in about half a day.

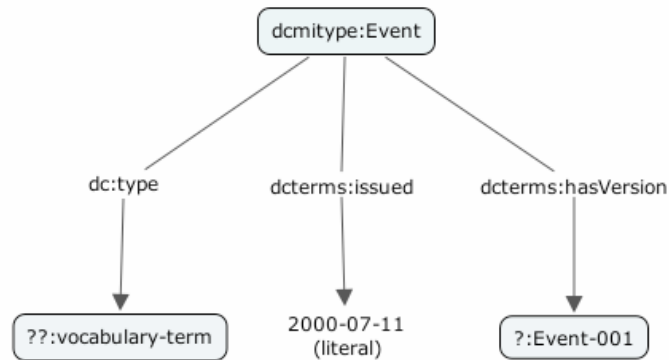
## If you know about OWL and the Semantic Web.....

COE allows you to view, edit and compose OWL ontologies without needing to know anything about, or ever look at, XML or RDF. Just *looking* at OWL ontologies with COE is often very illuminating. The concept map GUI can be learned by SWeb newbies in about half a day.

**BUT** composing good ontologies still requires some care.

COE provides access to concept clusters from published SWeb ontologies, constructed using the MVP-CA algorithm. This provides a unique perspective on the global conceptual structure of the network.

RDF is basically a set of triples, and so is a Cmap...



... but OWL is a more complicated language

```

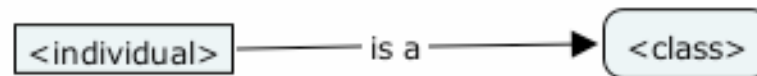
<owl:Class rdf:ID="Choice">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ControlConstruct"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#components"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#ProcessComponentBag"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
  
```



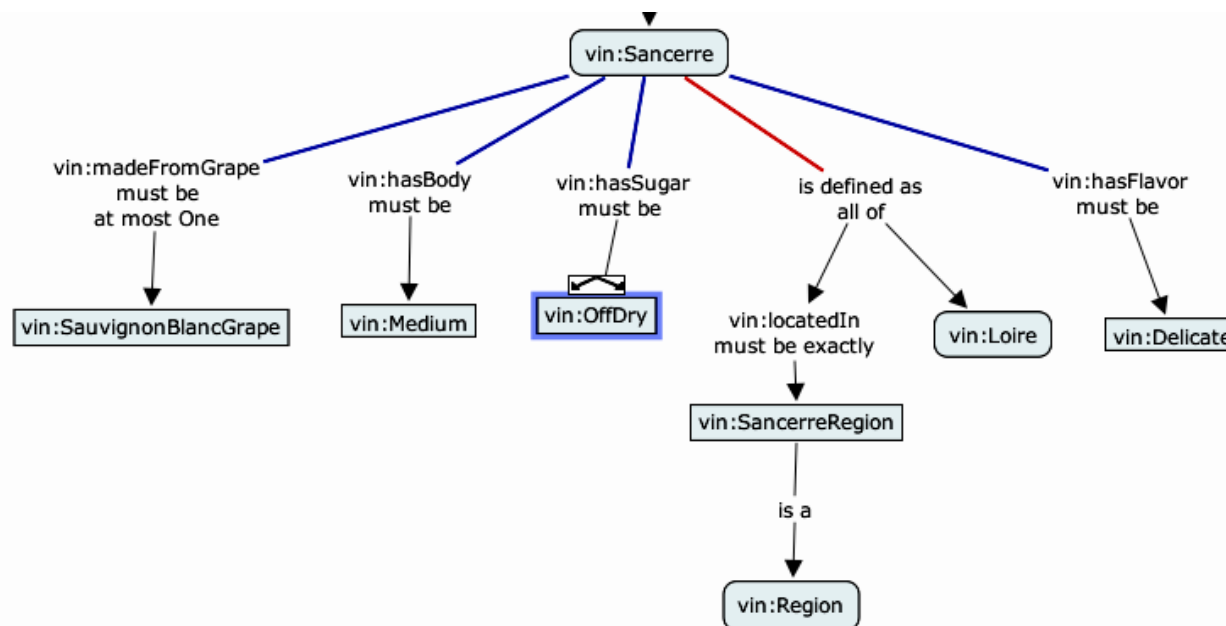
# Putting OWL syntax into a Concept map

## Basic ideas:

Individuals and classes are nodes, properties are links.

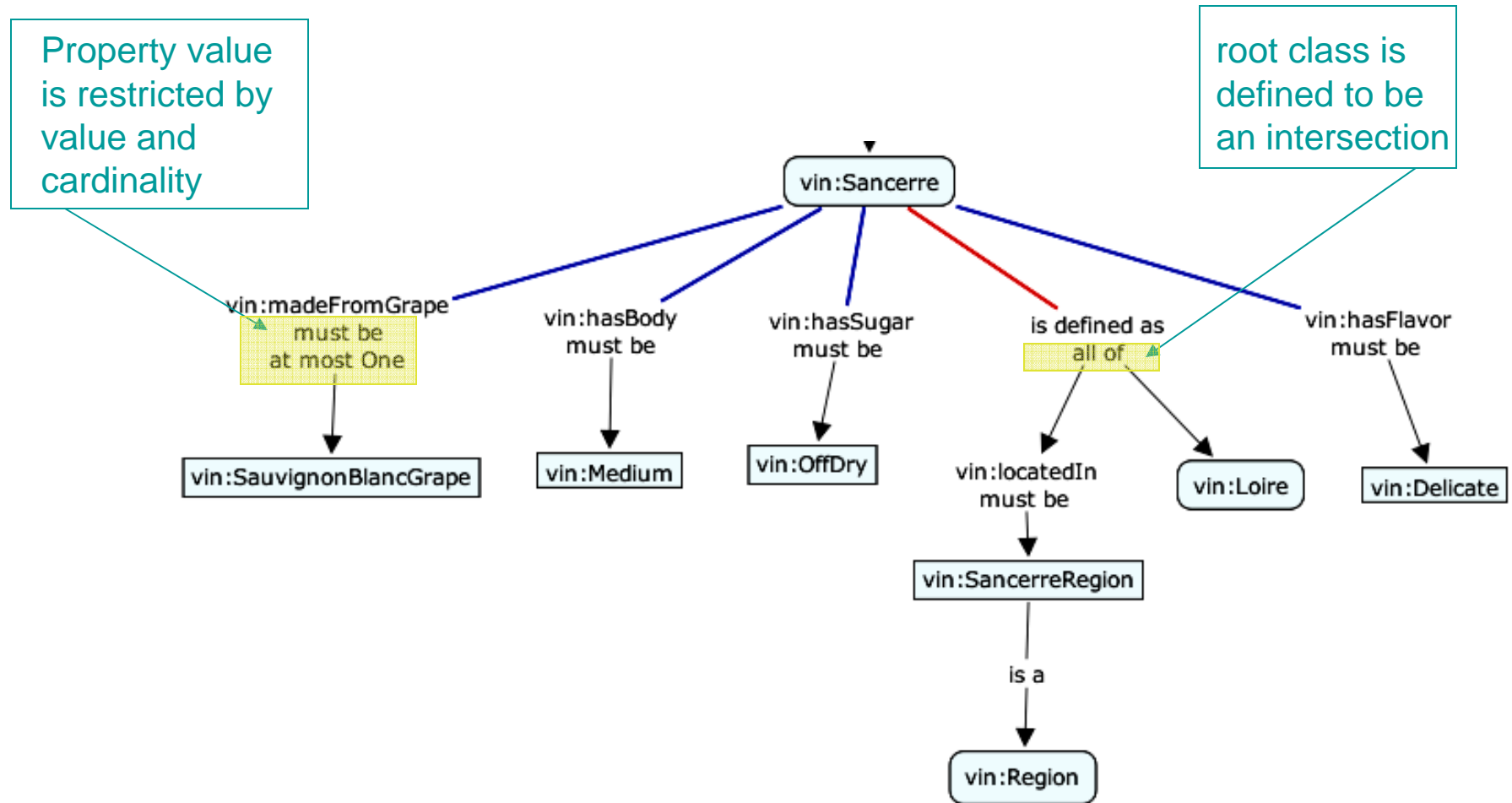


*Sub-* links are blue, exact definitions are red:



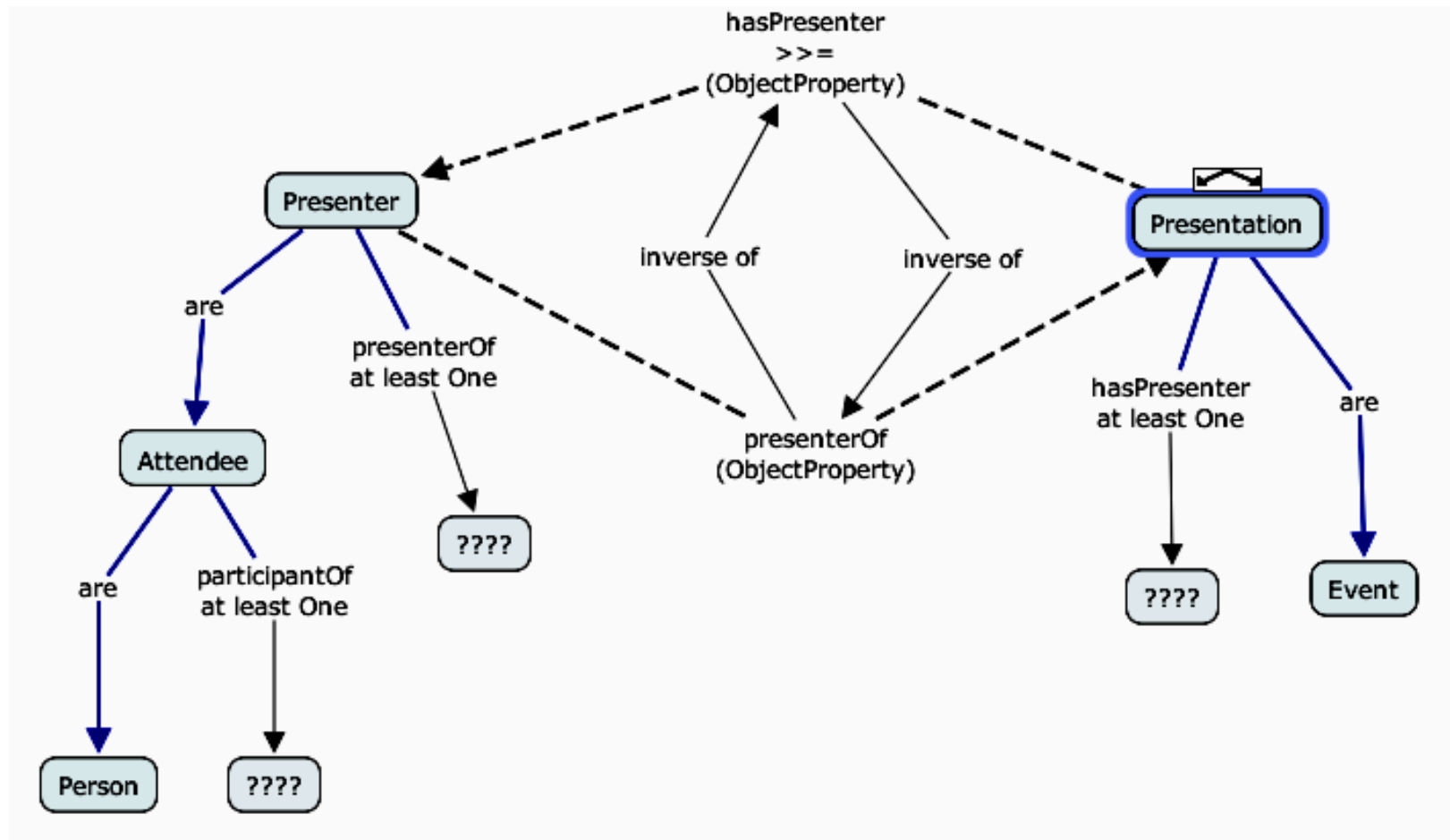
# Putting OWL syntax into a Concept map

Restrictions on properties are indicated by textual labels under the property name.



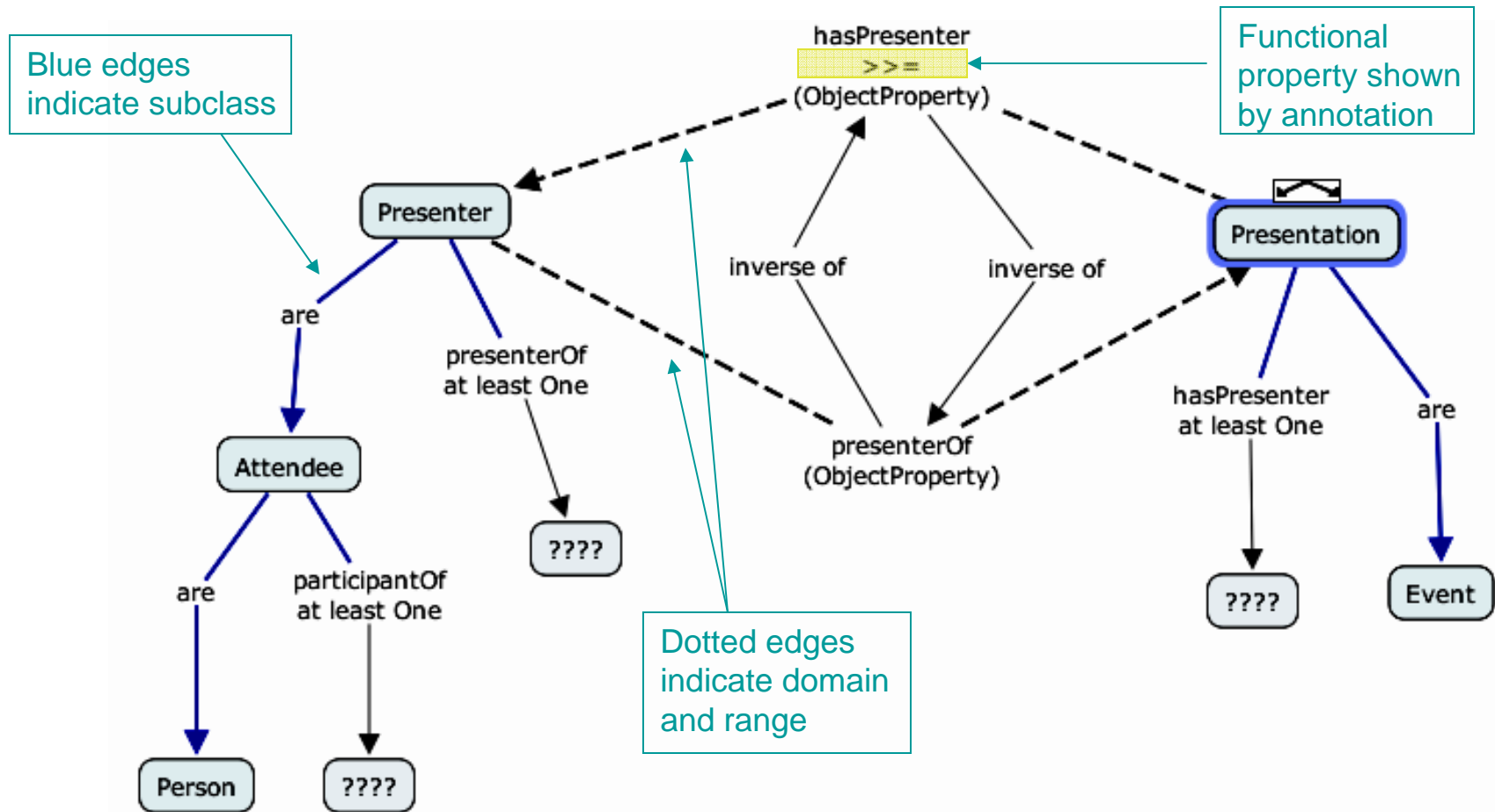
## Putting OWL syntax into a Concept map

Domains and ranges, properties of properties and other exotica are indicated using graphical conventions:



# Putting OWL syntax into a Concept map

Domains and ranges, properties of properties and other exotica are indicated using graphical conventions:



# Using COE

COE can be used as:

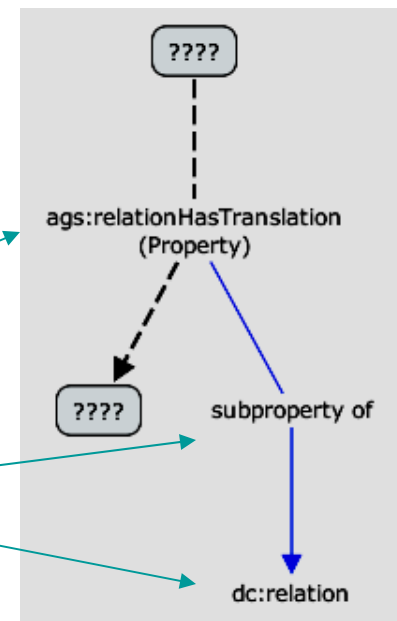
- an ontology viewer
- an ontology editor
- a concept search engine.

Integrating these abilities into one user interface is important for SWeb-style ontology development, where *re-use of concepts from existing ontologies on the network* is a basic mechanism for achieving interoperability.

ags: = <http://www.agls.gov.au/rdf/1.2/agls.rdfavailability>

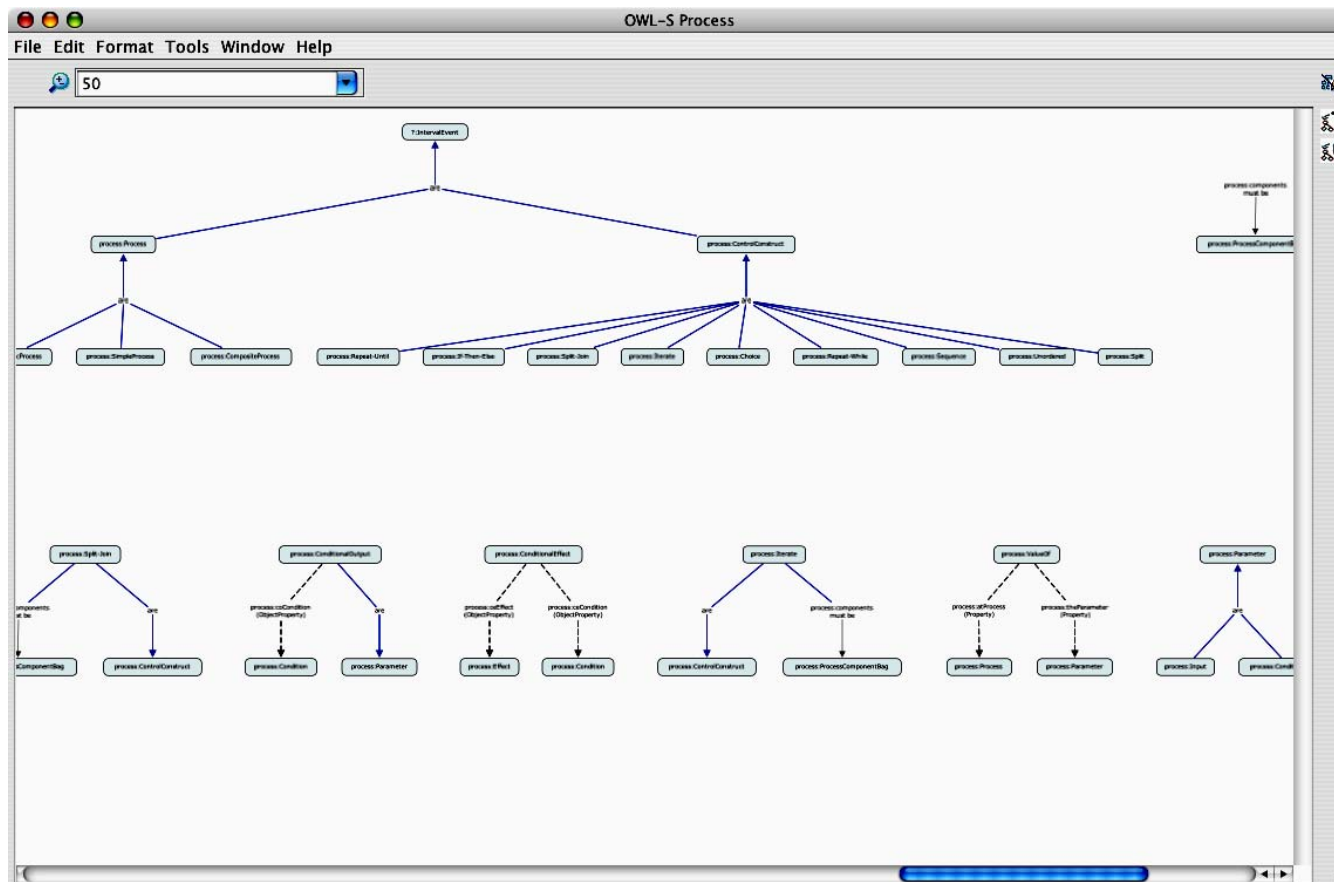
dc: = <http://purl.org/dc/elements/1.1>

owl: = <http://www.w3.org/2002/07/owl>



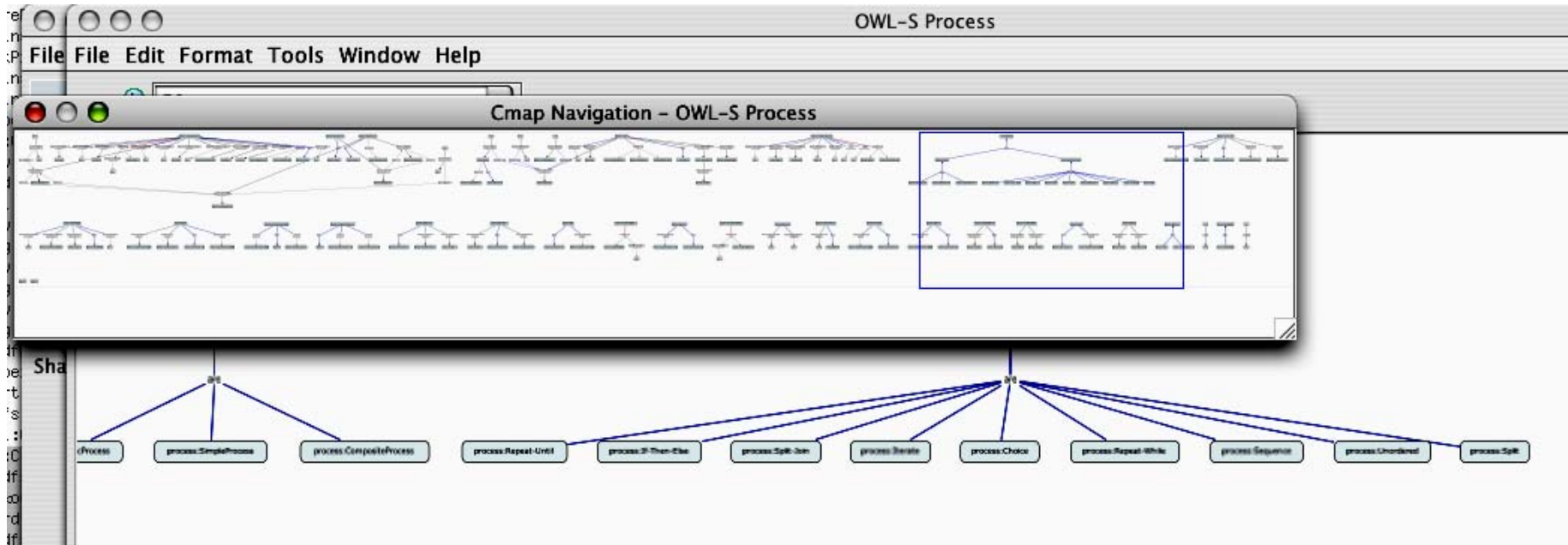
# Using COE as an ontology viewer

COE imports OWL/RDFS/RDF ontologies from XML files (or URIs using http) and displays them as a new concept map. Layout is automatic. Stored ontology Cmaps can be modified and archived using Cmap Tools.



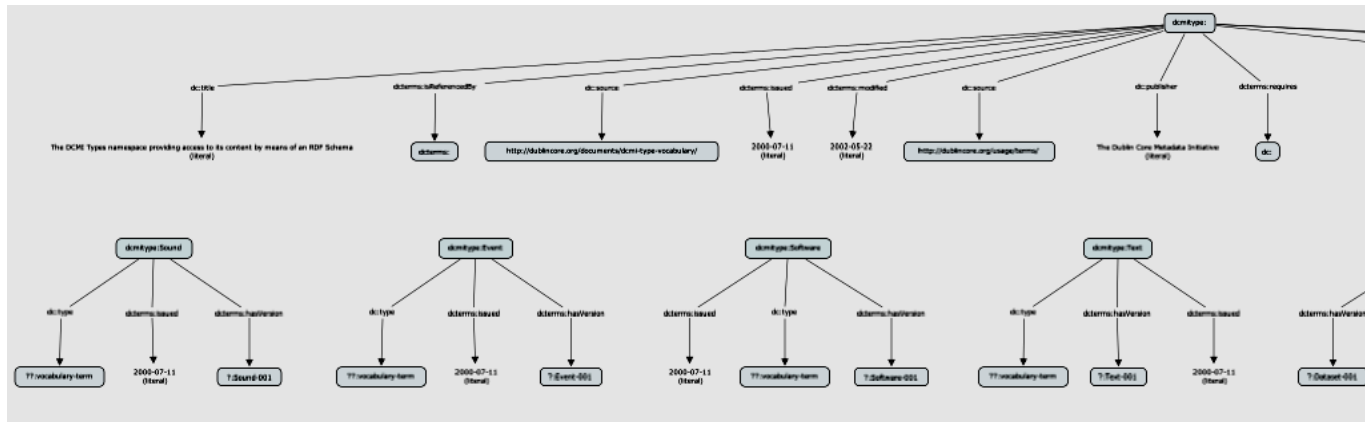
# Using COE as an ontology viewer

The navigation tool provides a useful overview of a large ontology



# Using COE as an ontology viewer

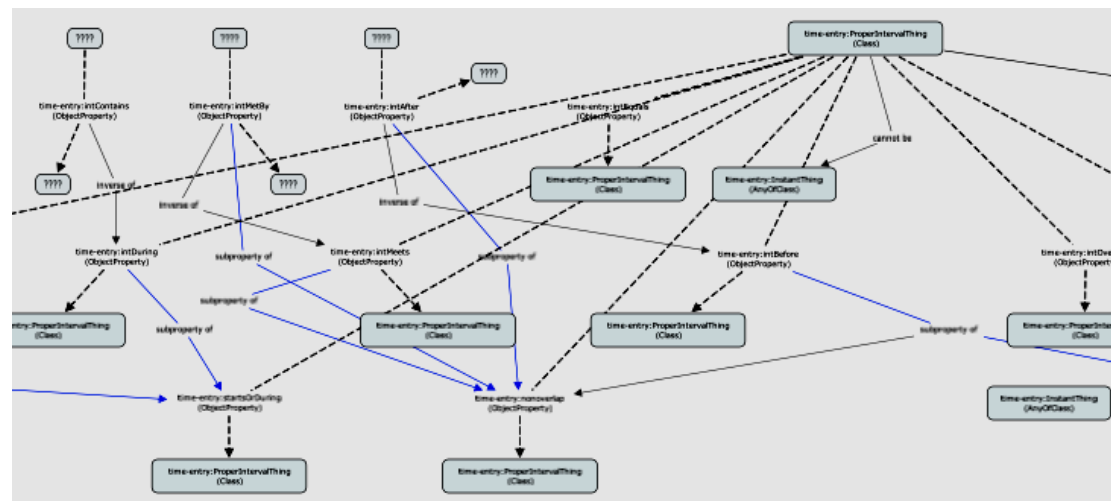
The COE view provides a rapid visual assessment of ontology content and 'style', and helps locate the most salient concepts.



No color, no long labels; mostly plain RDF

Lots of dashed lines: mostly about properties.

Nodes with large convergence are generic domains and ranges.

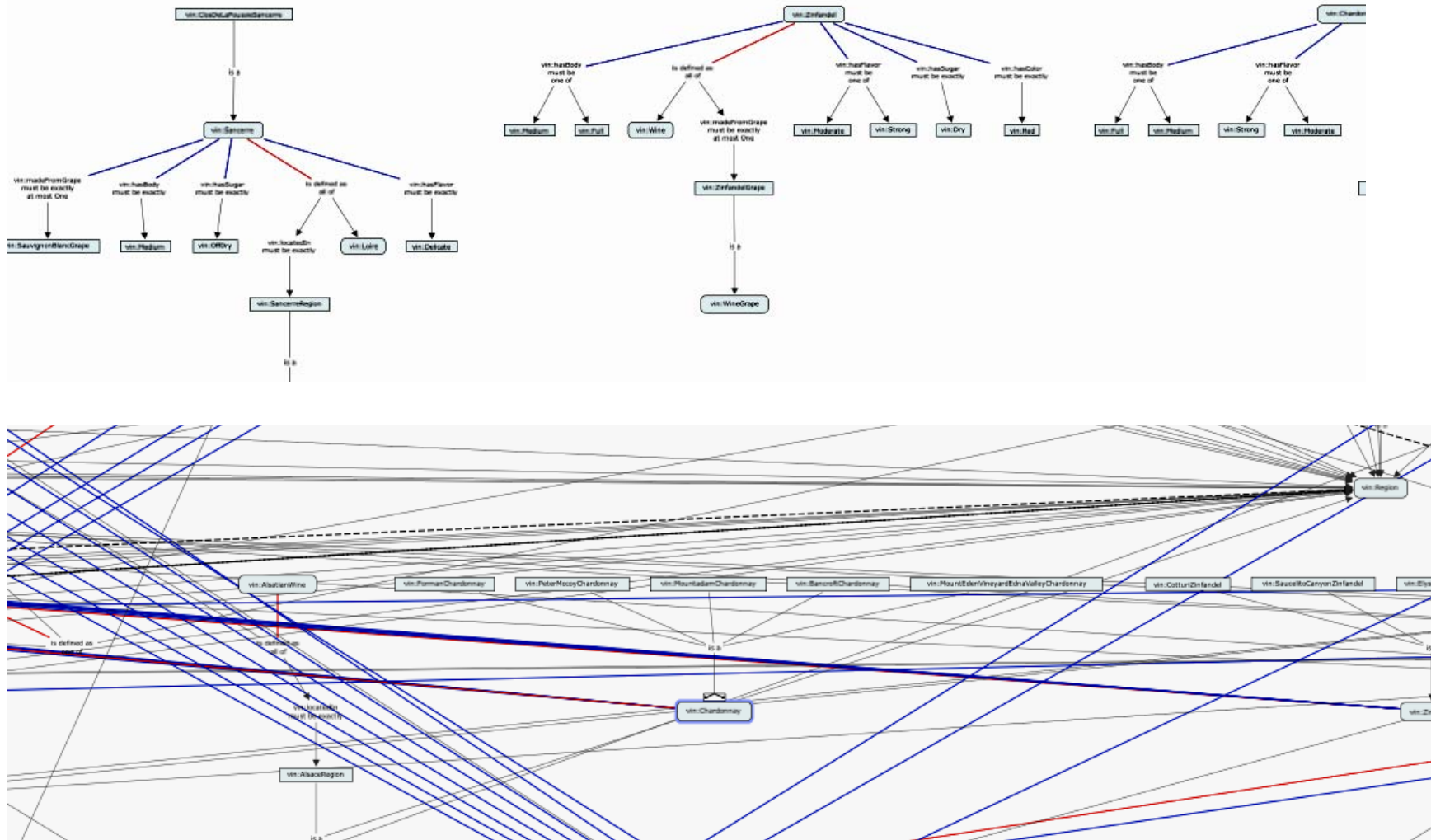






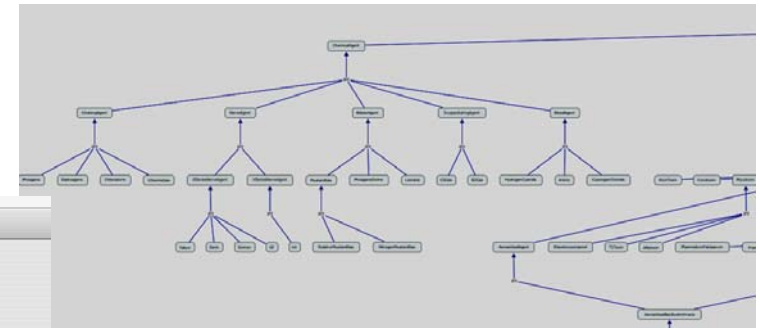
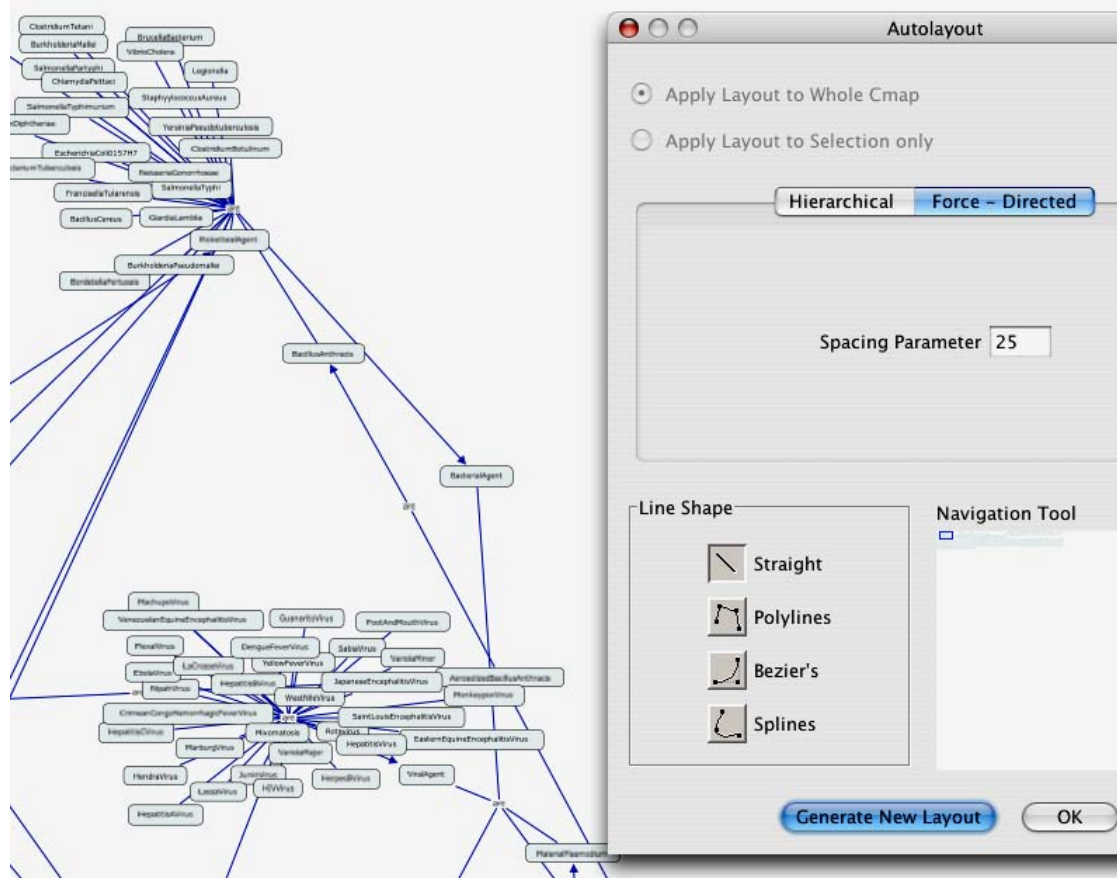
# Using COE as an ontology viewer

COE allows the same concept node to be copied to several places on the map surface, reducing clutter.



# Using COE as an ontology viewer

The auto-layout can be tuned to user preferences, and layout can be adjusted manually, and finished Cmaps archived for reference.



## Using COE as an ontology editor

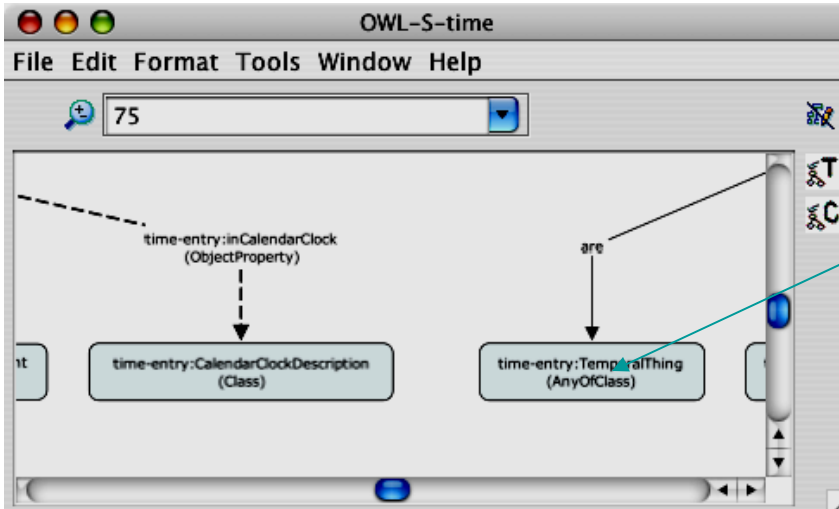
COE provides easy and intuitive cut-and-paste copying from existing Cmaps into new ones being composed. The Cmap Tools click-and-drag interface allows rapid editing and joining of Cmap fragments. COE extends this with templates and drop-down link menus, and has improved facilities for searching for concept names in and among ontology Cmaps.

COE will export any Cmap into OWL. The output is always syntactically correct OWL/XML: any parts that are not translatable into OWL are ignored. This allows 'formal' and 'informal' content to be mixed in the same Cmap, and informal Cmaps to be gradually extended by adding OWL content. Re-importing the OWL file as a Cmap gives a visual 'reality check' on the state of the formalization.

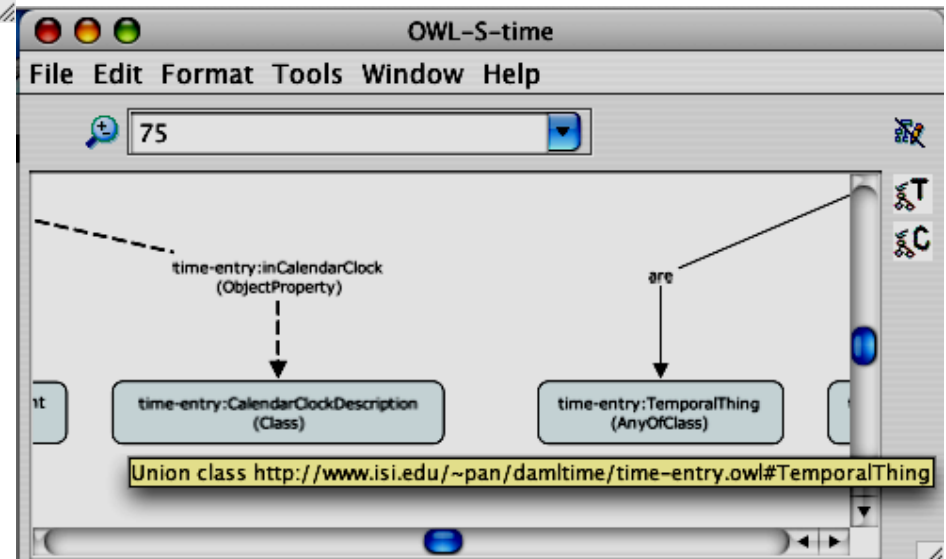
Complex ontologies may be composed in parts as separate Cmaps in an ontology folder, and COE will export them as a single OWL file.

# COE user interface additions

Hovering over a node shows you the full Web URI name of the concept in that node

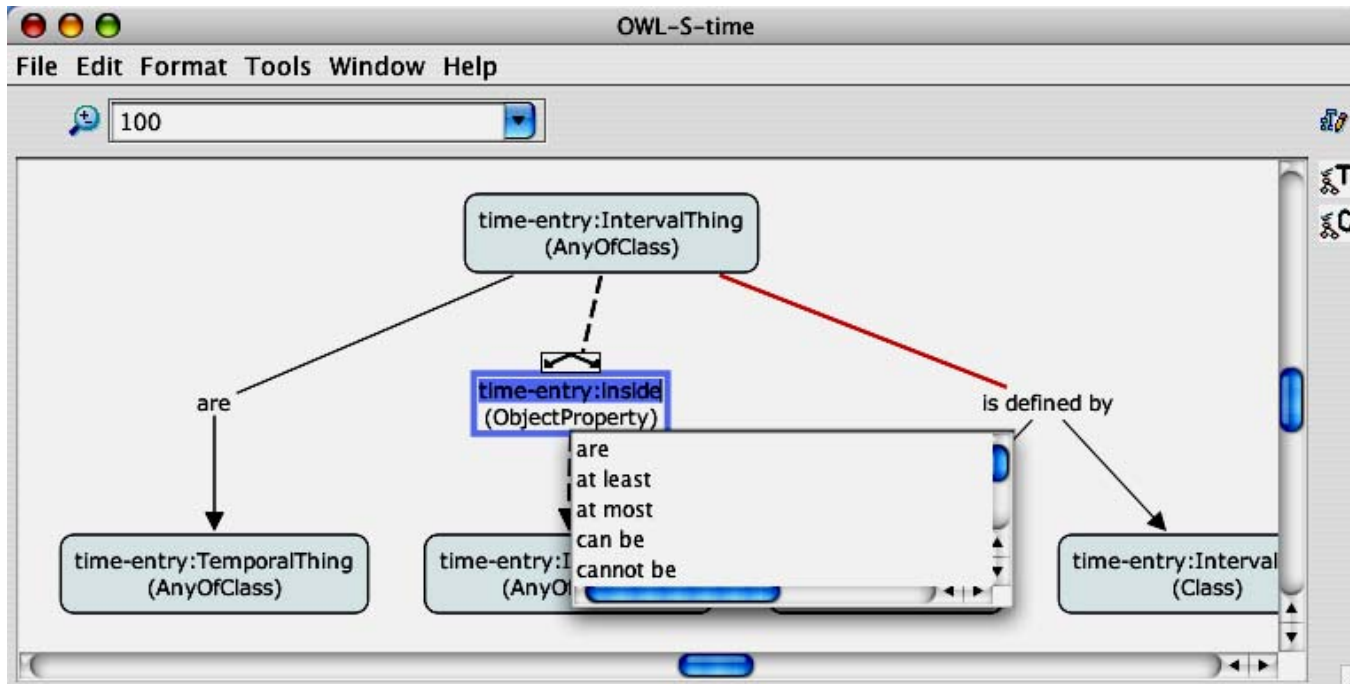


The mouse is here, but you can't see it in this screen shot.



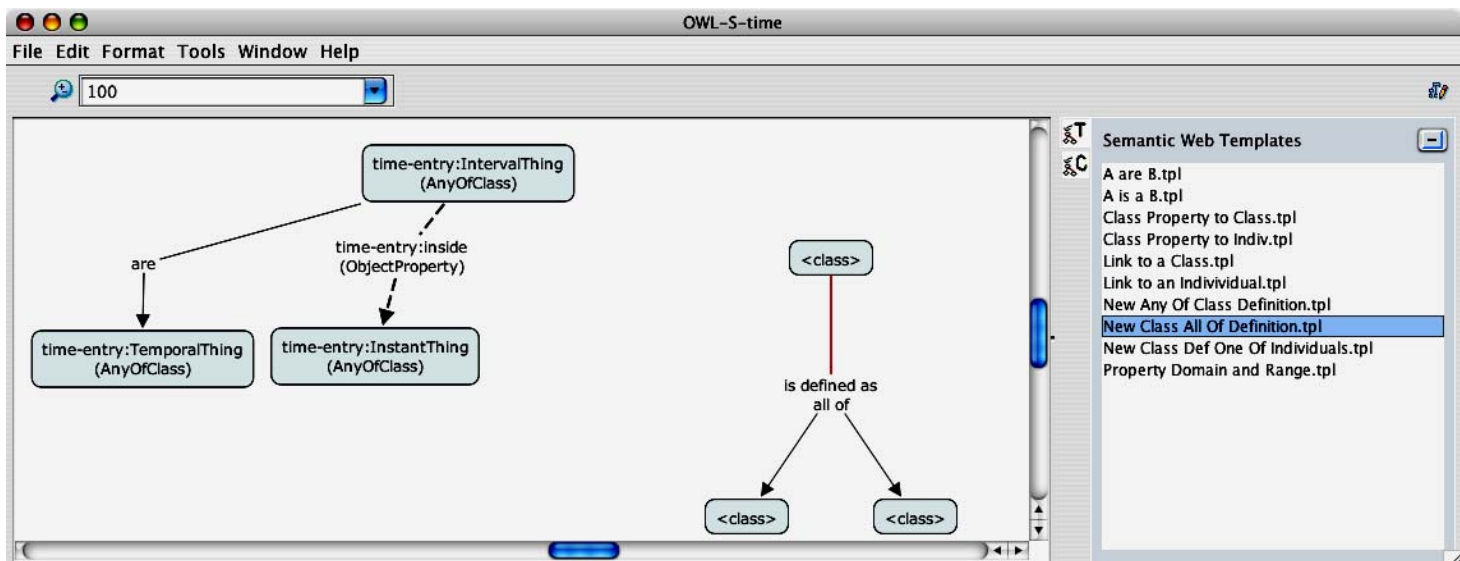
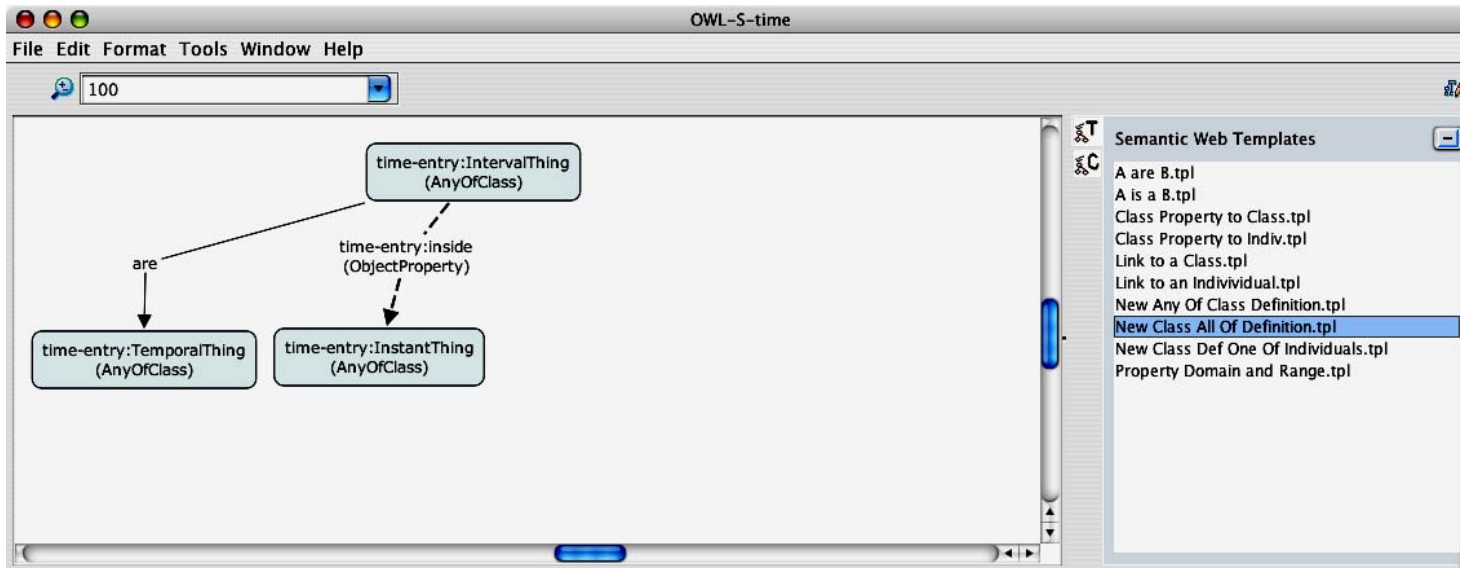
# COE user interface additions

Selecting a link label opens a drop-down menu of link names and components



# COE user interface additions

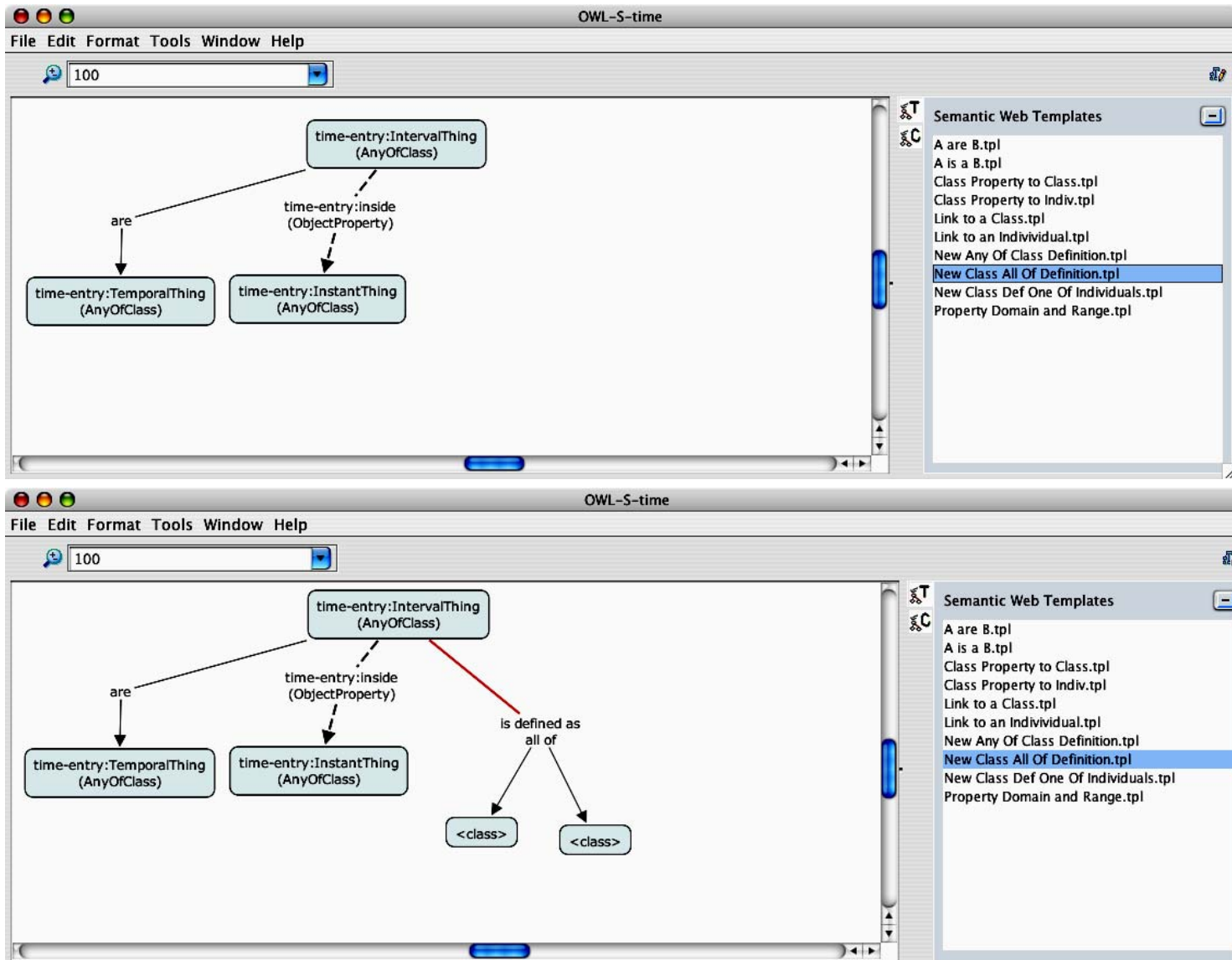
Templates of commonly used COE structures can be dragged into a Cmap





# COE user interface additions

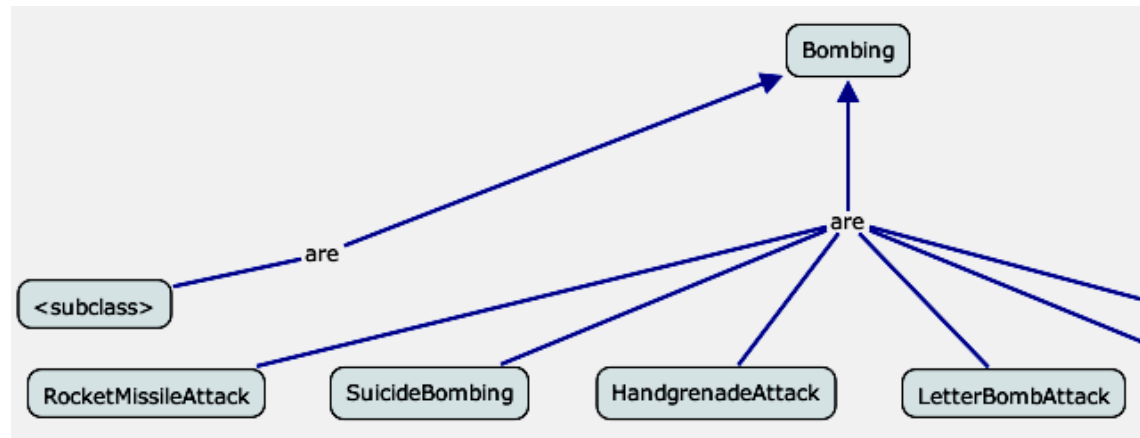
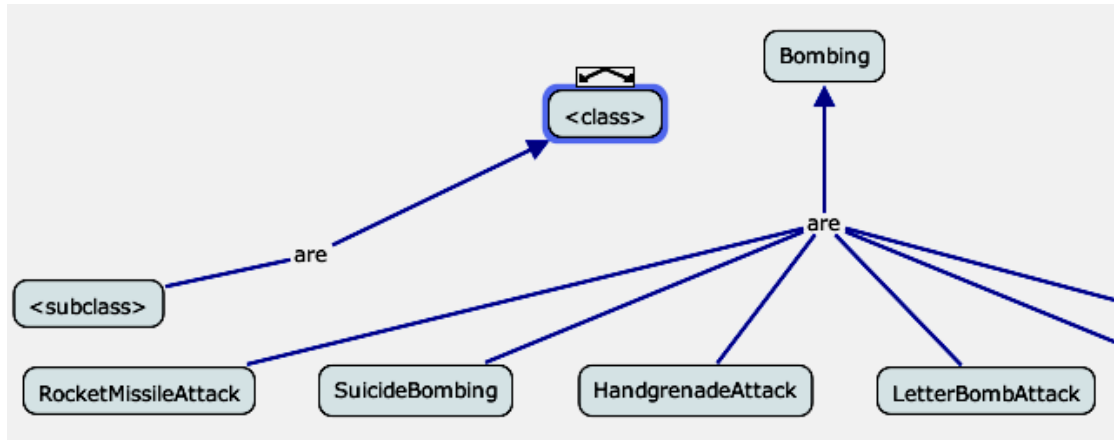
Templates of commonly used COE structures can be dragged onto a node in a Cmap





# COE user interface additions

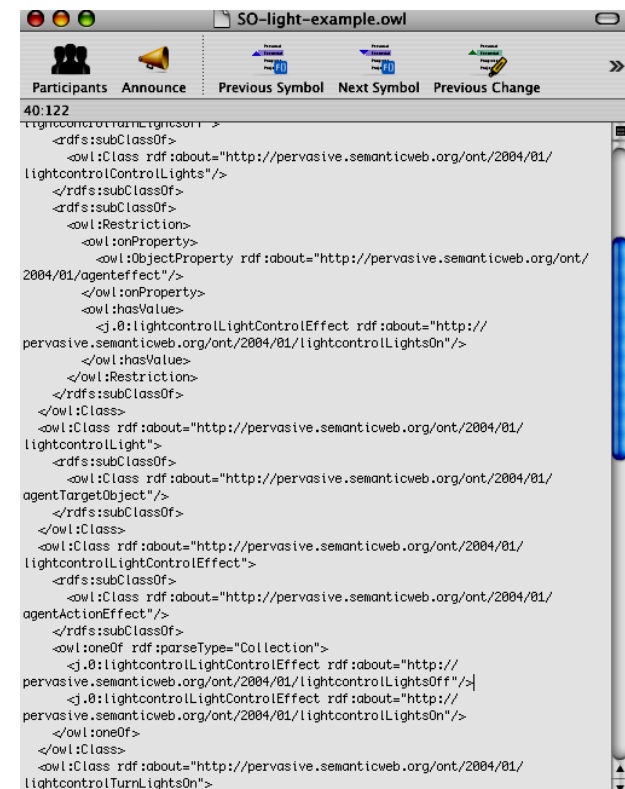
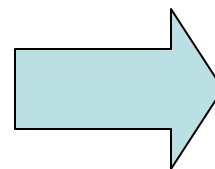
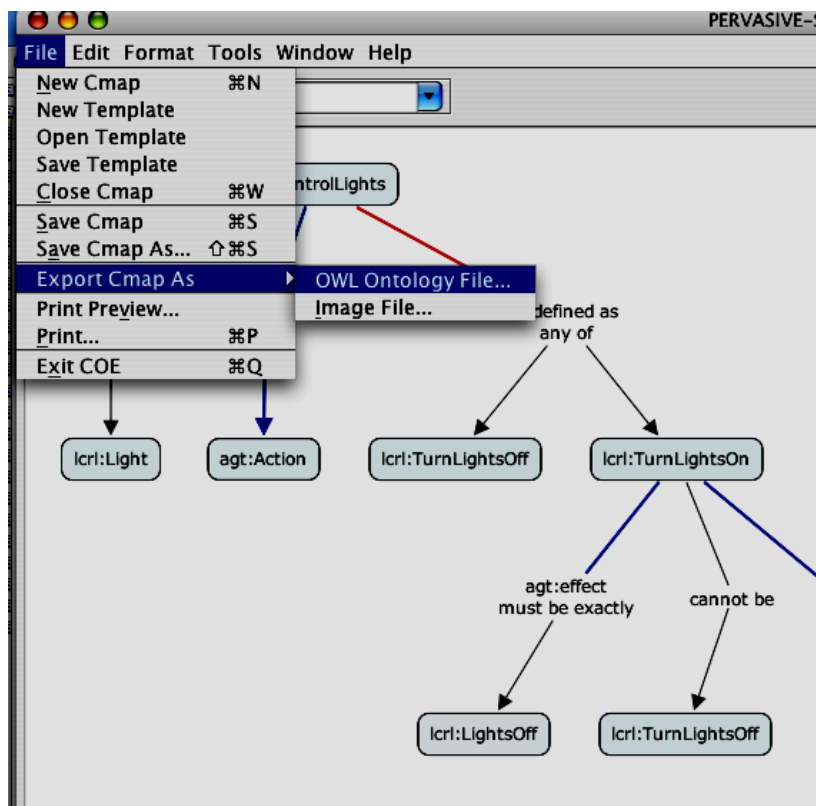
Nodes can be easily merged with a control-shift drop



# Using COE as an ontology editor

COE will export any Cmap into OWL. The output is always syntactically correct OWL/XML: any parts that are not translatable into OWL are ignored. This allows 'formal' and 'informal' content to be mixed in the same Cmap, and informal Cmaps to be gradually extended by adding OWL content. Re-importing the OWL file as a Cmap gives a visual 'reality check' on the state of the formalization.

Complex ontologies may be composed in parts as separate Cmaps in an ontology folder, and COE will export them as a single OWL file.



```
40:122
<lightcontrolTurnLightsOn >
  <rdf:subClassOf>
    <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
lightcontrolControlLights"/>
  </rdf:subClassOf>
  <rdf:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://pervasive.semanticweb.org/ont/
2004/01/agenteffect"/>
      </owl:onProperty>
      <owl:hasValue>
        <j.0:lightcontrolLightControlEffect rdf:about="http://
pervasive.semanticweb.org/ont/2004/01/lightcontrolLightsOn"/>
      </owl:hasValue>
      <owl:Restriction>
        <rdf:subClassOf>
          <owl:Class>
            <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
lightcontrolLight">
              <rdf:subClassOf>
                <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
agentTargetObject"/>
              </rdf:subClassOf>
            </owl:Class>
          </owl:Class>
          <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
lightcontrolLightControlEffect">
            <rdf:subClassOf>
              <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
agentActionEffect"/>
            </rdf:subClassOf>
            <owl:oneOf rdf:parseType="Collection">
              <j.0:lightcontrolLightControlEffect rdf:about="http://
pervasive.semanticweb.org/ont/2004/01/lightcontrolLightsOff"/>
              <j.0:lightcontrolLightControlEffect rdf:about="http://
pervasive.semanticweb.org/ont/2004/01/lightcontrolLightsOn"/>
            </owl:oneOf>
          </owl:Class>
        </owl:Class>
      </owl:Class>
    </owl:Class>
  </owl:Class>
  <owl:Class rdf:about="http://pervasive.semanticweb.org/ont/2004/01/
lightcontrolTurnLightsOn">
```

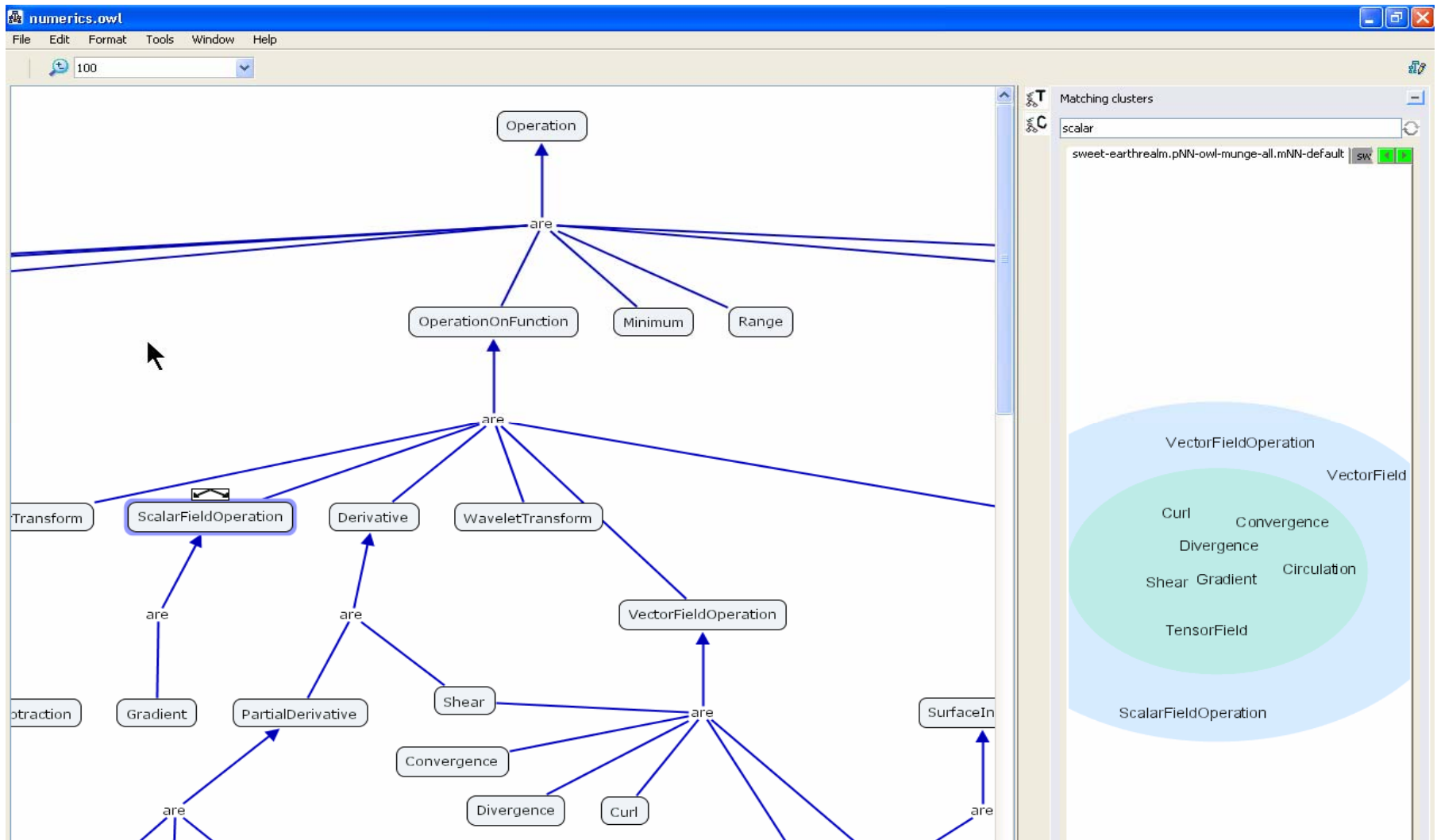
## Using COE as a concept search engine

**Distributed syndication** depends on composers of new content having easy access to concepts, and enough information to enable them to make rational choices between existing concepts or inventing their own.

COE provides a simple text search through archived Concept Maps in the local files and the IHMC Public Ontology Server.

The COE architecture provides for a variety of more sophisticated tools for displaying concept information. A graphical interface to the MVP-CA clustering system developed by Pragati is incorporated into the current version of the software, to display multi-viewpoint clusters of concepts in OWL ontologies currently available on the Semantic Web. Concepts may be used to identify and retrieve related ontologies, and be incorporated into new Cmaps.

# Context Exposition in COE/MVP-CA



March 14-15, 2005



# Concepts related to “*Operation*”

