Ontology comprehension

Johann Bergh

November 9, 2009

Contents

1 Definitions

`Protégé:` Protégé is an advanced software ontology editing tool.
`Taxonomy:` A taxonomy is a particular classification arranged in a hierarchical structure.
`Asserted taxonomy:` Explicitly defined taxonomy.
`Inferred taxonomy:` Resulting taxonomy after a mathematical reasoner has been run on the asserted taxonomy.
`Asserted graph:` Graph containing the asserted taxonomy and existential relationships emanating from concepts in the asserted taxonomy.
`Inferred graph:` Graph containing the inferred taxonomy and existential relationships emanating from concepts in the inferred taxonomy.
`TBox:` List of terminological axioms.
`ABox:` List of assertional axioms.
`Subsumption relationship:` A relationship between concepts. Illustrations of such relationships are usually displayed in tree-like structures as *is-a* relationships, meaning that children are more specific than their parents. A taxonomy can be described as a set of subsumption relationships.
`Existential relationship:` A relationship between concepts, where a concept is linked to another concept, via a defined relationship, at least once.

2 Introduction

Large ontologies become complex and difficult to understand. This idea is supported by Bauer([1], [2]) and it lead him to write a model exploration software plug-in for Protégé. The software tool, called *SuperModel*, builds models from root concepts in ontologies and visualises it to the user. The user can also manipulate the model and test its consequent satisfiability.

The aim of the software tool written by Bauer was to aid users in the understanding of ontologies. We endeavour to take the idea of ontology understanding further, but extend it as a field within ontologies called *ontology comprehension*.

The notion of ontology comprehension is illustrated by the creation of a new tab in Protégé. This tab will contain all the Protégé plug-ins that aid the user in understanding ontologies. Therefore, we moved SuperModel from its previous location to the new ontology comprehension tab. Furthermore, we created another plug-in called *PathViz* that will also aid users in understanding ontologies better. More details on PathViz will follow in Section 4. To summarize, we have the new ontology comprehension tab with two plug-ins on the tab, namely SuperModel and PathViz. Figure 1 illustrates this situation. Ultimately, the idea is that software developers will write additional plug-ins that will aid ontology understanding. These plug-ins could be added to the ontology comprehension tab. The result is that there will be several tools to the disposal of the users to aid them in better understanding ontologies.

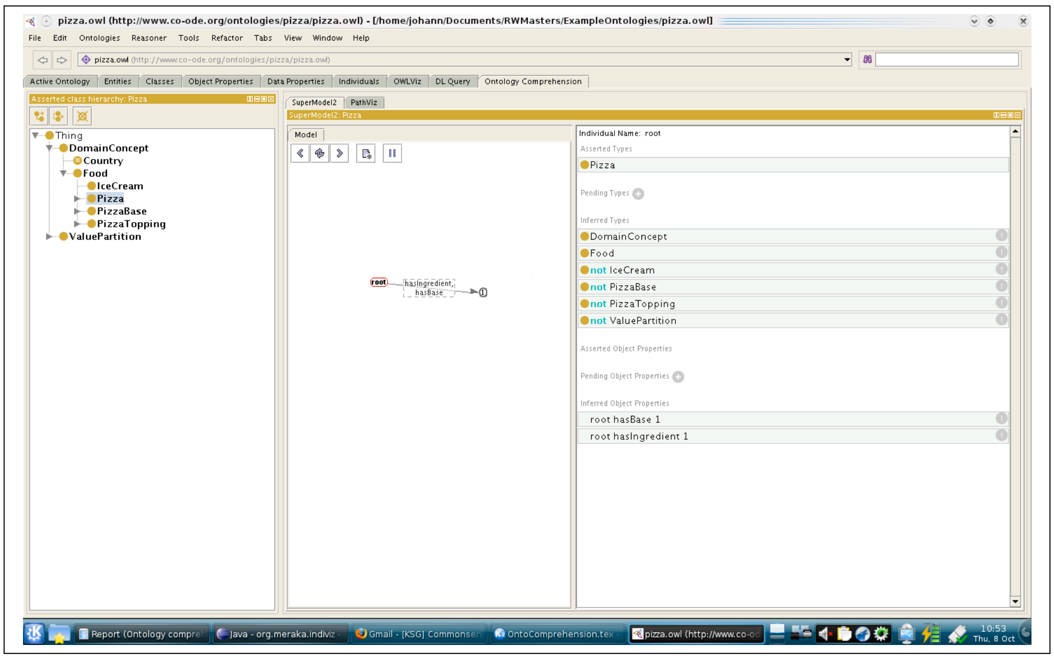

Figure 1: *Ontology comprehension* tab in Protégé

Ontology comprehension tools will often use visualisation techniques to facilitate the understanding process, but are not limited to it.

The remainder of this document is divided into several sections. In Section 3 we explain the significance of ontology comprehension. In addition, we explain how SuperModel and PathViz relate to the field of ontology comprehension. In Section 4 we proceed to discuss the Protégé plug-in that we developed, PathViz. We look at additional features of PathViz in Section 5. A discussion on possible future work can be found in Section 6. In Section 7 and 8 we illustrate some formal representations of the aspects that we dealt with in this document, before we conclude in Section 9.

3 Significance of ontology comprehension, SuperModel and PathViz

Large ontologies become complex and difficult to understand([1], [2]). Consequently, we question how we can ease the process of understanding ontologies, and especially larger ontologies, for the user (of the ontology). The notion of ontology comprehension aims to answer this question.

Within the framework of ontology comprehension, there are many ways to ease the process of understanding ontologies. SuperModel and PathViz are examples of these and therefore they fall within the framework of ontology comprehension. Henceforth, we propose that SuperModel and PathViz are *elements of ontology comprehension*. Ontologies in themselves comprise many aspects such as concepts, relations and instances. In the same way, we argue that different elements of ontology comprehension can address the understanding of the different aspects of ontologies. For example, SuperModel builds models, that serve as practical examples, of an ontology to give users an idea of how the ontology can be used. PathViz, on the other hand, helps users to understand how different concepts in an ontology are related to each other.

SuperModel and the details surrounding model exploration have been documented in detail. The interested reader is referred to the work of Bauer([1], [2]).

As stated before, we propose that PathViz is an element of ontology comprehension. PathViz addresses two important questions within the framework of ontology comprehension. Firstly it looks at how different concepts are related to each other in an ontology. Secondly it shows what influence reasoning processes, performed on ontologies, have on the relationships between different concepts in ontologies. To highlight this, we define the notion of the *path simplicity ratio*. A high value for the ratio indicates that the reasoning process has little effect on the relationship between different concepts in the ontology. Similarly, a low value indicates that the reasoning process has a substantial effect. More details are provided in Section 8.

4 PathViz

PathViz is a Protégé plug-in that uses a step based wizard approach to aid users in understanding ontologies. Graphs of both the asserted and inferred taxonomies are created, subsequently referred to as the asserted graph and the inferred graph (more details can be found in Section 7). In this section, it is sufficient to know that all the concepts in the ontology are contained in the asserted and inferred graphs.

When using PathViz, the user selects two concepts in the ontology (Figure 2). PathViz computes a set of paths between the two selected concepts for both the asserted and inferred graphs. These paths are then rendered visually to the user. This situation is depicted in Figure 3. The user can iterate through both sets of paths (obtained from asserted and inferred graphs) and visually compare them. Each rendered path has a status label to indicate whether that specific path can also be found in the path list from the other graph. For example, if we are looking at one of the paths in the set obtained from the inferred graph, we will be able to see if there is a similar path in the set of paths obtained from the asserted graph.

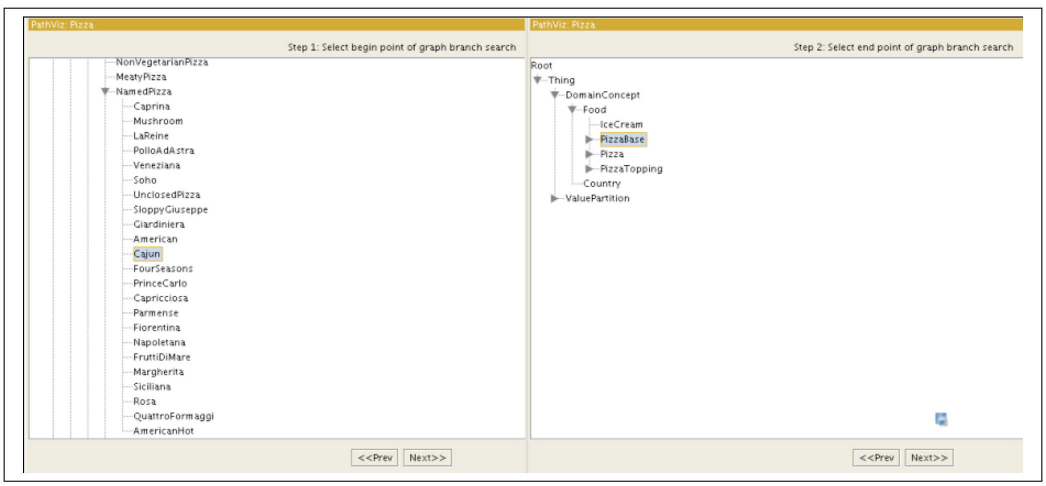

Figure 2: First two steps of the *PathViz* Protégé plugin

For clarity, we now discuss the details of the construction of the inferred and asserted graphs. These graphs are constructed from two types of relationships in the ontology. Firstly we explore the subsumption relationship. This part of the graph construction takes the taxonomy of the ontology and places it in a graph structure (concepts are linked to each other with *is-a* relationships). Secondly we explore existential relationships in the ontology. Existential relationships that form part of concepts in the ontology are also added to our graph structure. Note that this process is followed for both

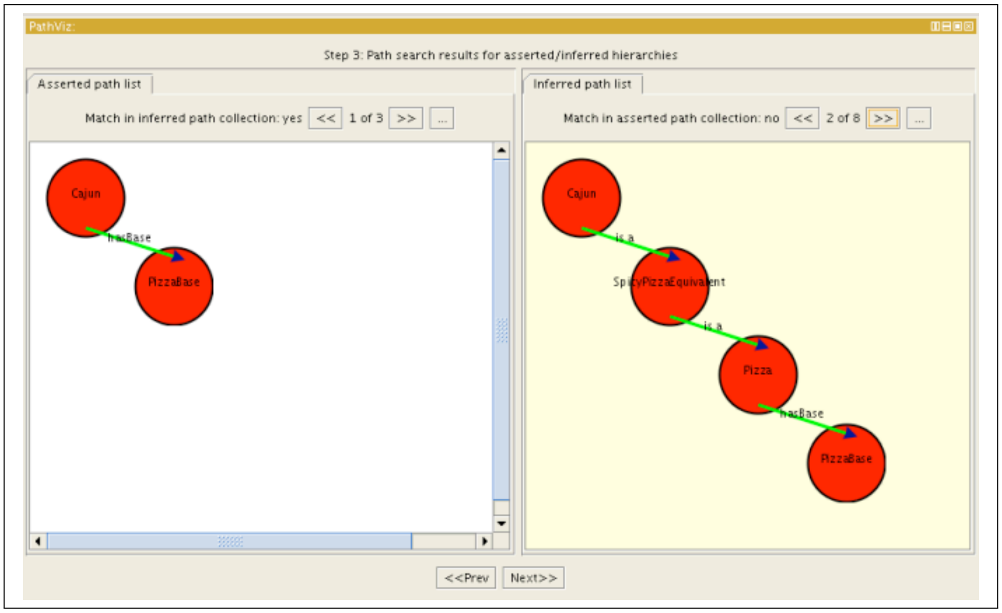

Figure 3: *PathViz* search results

asserted and inferred taxonomies of the ontology. Please refer to Section 7 of this document for a formal representation of this process.

It is beyond the scope of this discussion to do a detailed comparison between PathViz and SuperModel, but for completeness we highlight the most prominent differences, since the conception of PathViz was largely inspired by SuperModel. PathViz provides an advantage that a tool like SuperModel does not address, namely immediate path visualisation. SuperModel works on the principle of iteratively expanding a model from a root concept, but if the user has a goal concept in mind it might take many iterations from the root concept before this goal concept is found, or it may not be found at all; PathViz, on the other hand, is more goal driven in the sense that a root and a goal concept are stated upfront and then a set of all the paths between these concepts are generated. Each path in the sets of paths generated by PathViz is not a model of the ontology, but simply a branch of the asserted or inferred graphs. The elements (vertices) of these graphs are concepts in the ontology. SuperModel, on the other hand, generates a model of the ontology upfront from a single concept, called the root concept. The elements within this model are, of course, individuals (instances of concepts). However, from each path in the sets of paths generated by PathViz, it is also possible to generate a model of the ontology. In contrast to SuperModel, this will happen as a final step of the PathViz process. More formal details can be found in Section 8. Furthermore, SuperModel is beneficial because the user can iteratively and interactively manipulate a model of the ontology and verify its consequent satisfiability. The user cannot manipulate paths in the sets of paths generated by PathViz, and such a course of action would serve no purpose. Models generated by PathViz could be passed on to SuperModel for manipulation by the user. However, the plausibility of such a notion should be further investigated. Readers should note that the generation of models by PathViz has not been implemented in the current version of the software, but is possible in theory.

5 Other features

- PathViz allows users to export the entire asserted and/or inferred graph to XML format. Individual paths in the resulting path sets can also be exported to an XML file.

6 Future work

There is room for expanding and improving the PathViz plug-in in the future. Some possible improvements will be dealt with briefly:

- Applying filters before the construction of the asserted and inferred graphs might be useful. For example, the asserted and inferred graphs could be constructed by excluding some or all taxonomic (*is-a*) relationships or by excluding some or all existential relationships.

- Before the construction of the asserted and inferred graphs, axioms can be added to the ontology temporarily to observe the effects thereof.

- PathViz involves the construction and traversal of graphs, amongst others. An investigation into the time complexity of these elements will yield useful information. For example, this information can be used to determine the effectiveness of PathViz for larger ontologies.

- The asserted and inferred graphs in PathViz are directed graphs that are weakly connected. This means that one will be able to reach any vertex in the graph from the head node (vertex) of the graph, but one will not necessarily be able to reach any vertex from a vertex other than the head node (vertex); the head node being the top node of the asserted and inferred taxonomies that was used to construct the asserted and inferred graphs respectively. In the light of this information we would like to asses the reachability of the asserted and inferred graphs. In other words, under which circumstances it is possible to find a path between two arbitrary vertices in the graphs.

- Generating and visualising a model of the ontology from an arbitrary path in the set of paths of the asserted and inferred graphs.

- Lastly, it might be useful to investigate whether other ontological concepts such as transitivity can be incorporated in the graphs.

7 Formal representation of PathViz process

In this section we present the PathViz process formally and make some important remarks. We observe an ontology O, where T is the *TBox* and A is the *ABox*:

$$O := < T, A > \ .$$

Let C and R refer to a set of concepts and roles respectively in the ontology. Axioms in a *TBox* are constructed by using C, R and boolean constructors.

When we execute a tableaux reasoner B over the ontology O, we obtain a new set of terminological axioms, T':

$$T' = B(O) \ .$$

Because tableaux reasoners are monotonic it is easy to see that T is a subset of T':

$$T \subseteq T' \ .$$

A directed graph is an ordered pair

$$G := (V, E) \ ,$$

where V is a set of nodes (vertices) and E is a set of ordered pairs of nodes (vertices).

From here we use T to construct the asserted graph, $G(V, E)$, where

$$V = C \text{ and}$$

$$E = Q(V) \ .$$

Here, Q is a function that returns a set of objects that links two concepts. This set is obtained by extracting the asserted taxonomy and the existential relationships therein, from the ontology.

Similarly, the inferred graph, $G'(V', E')$, is constructed from T' where

$$V' = C \text{ and}$$

$$E' = Q'(V') \ .$$

If we now choose an arbitrary v_r as a root concept and an arbitrary v_g as goal concept from C, we define a function F to compute a set of paths P between v_r and v_g in graph G_i:

$$P := F(v_r, v_g, G_i) \ .$$

For a fixed v_r and v_g we proceed to compute a set of paths P_a and P_i for the asserted and inferred graphs respectively, where

$$P_a = F(v_r, v_g, G(V, E)) \text{ and}$$

$$P_i = F(v_r, v_g, G'(V', E')) \ .$$

We conclude by using the sets P_a and P_i for further analyses.

8 Other formal observations

We observe that

$$\mid P_i \mid \geq \mid P_a \mid \ ,$$

meaning that the cardinality of the set of paths computed for the inferred graph will always be greater or equal to the cardinality of the set of paths computed for the asserted graph.

Thus, we proceed to define the path simplicity ratio, R_p, for the sets of paths, P_a and P_i, as follows

$$R_p := \frac{\mid P_a \mid}{\mid P_i \mid} \ .$$

It is then also easy to see that

$$R_p \in [0, 1] \ .$$

For a fixed v_r and v_g, we can extend this definition to

$$R_p(v_r, v_g) := \frac{| P_a |}{| P_i |} \ ,$$

where $R_p(v_r, v_g)$ is a more elaborate notation to make it clear that the path simplicity ratio was computed for paths where v_r was the root concept and v_g was the goal concept. A path simplicity ratio of one or close to one indicates that the reasoning process had little effect on the way v_r relates to v_g. We can also say that the relationship of v_r with v_g is *simple*. On the other hand, a path simplicity ratio of zero or close to zero means that the reasoning process had a substantial impact on the way v_r relates to v_g (relationship of v_r with v_g is *complicated*).

At this point, it is important to further observe that

$$R_p(v_r, v_g) \neq R_p(v_g, v_r) \ .$$

Some noteworthy deductions follow from this observation. If v_r has a complicated relationship with v_g it does not mean that v_g has a complicated relationship with v_r. It could be that v_g has a simple or non-existent relationship with v_r. Indeed, on closer inspection, it is easy to see that the asserted and inferred graphs used in the computation of sets of paths, P_a and P_i, would be completely different for $R_p(v_r, v_g)$ and $R_p(v_g, v_r)$ respectively. This notion is logicaly strengthened if one recalls that the edges in the asserted and inferred graphs are not bi-directional.

If, however, $R_p(v_r, v_g)$ and $R_p(v_g, v_r)$ both have low values, we state that v_r and v_g have a *reflexively complicated relationship*. Similarly, if $R_p(v_r, v_g)$ and $R_p(v_g, v_r)$ both have high values, v_r and v_g have a *reflexively simple relationship*.

Finally, we believe that it will be possible to devise a general algorithm to construct an ABox, A, from every path, p, in the sets of paths, P_a and P_i, such that A will be a model of the ontology O:

$$O \models A \ .$$

9 Conclusion

Large and complex ontologies introduce a huge learning curve for those who want to use it. This leaves us with the question of how we can make ontologies, and especially larger ontologies, easier and quicker to understand for the users thereof. In this context we defined the notion of ontology comprehension. Within ontology comprehension there are different elements that aid us to understand ontologies, such as SuperModel and PathViz. SuperModel is an existing software tool that exploits model exploration to aid ontology understanding. Pathviz is a more recent software tool that ameliorates ontology understanding.

We now proceed to discuss the usefulness of PathViz as an element of ontology comprehension. PathViz uses visualisation techniques to aid the understanding process. We argue that the mere fact that PathViz gives a complete visual representation of how two concepts in an ontology relate to each other, is beneficial to the user. The path simplicity ratio is an important concept that was born out of the PathViz process. This ratio gives us an indication of how the relationship between two concepts in an ontology changes when a reasoner has been executed on the ontology. The path simplicity ratio can potentially have many useful applications in future. One such example would be to use the ratio to devise the most effective strategy to teach an ontology to a novice user. Intuitively, we would start by looking for all concepts in the ontology that have a reflexively complicated relationship. We could then focus the user's attention on these concepts and use visualisation techniques to explain them to the user. The theory is that, if the user understands the most complex concepts in the ontology, then the ontology as a whole would also be understood. This theory can then be compared with other theories, such as focusing the user's attention on reflexively simple relationships or a hybrid of different theories.

Though we can use arguments to prove that an element of ontology comprehension, such as PathViz, is useful, we would ideally like to conduct tests or experiments that give us measurable results. The path simplicity ratio experiment as described above could be used, but seeing that it is largely a user study, it comes with some constraints. User studies are often very arduous and inaccurate. For a successful user study, we would also require a large group of participants. Finding such a group will not necessarily be easy, as ontologies in the context of computer science is currently an emerging field. Bauer [1] made use of a user study to measure the usefulness of SuperModel, but expressed some frustrations in the process. Optionally, the example in Bauer's user study can be reused and adapted for PathViz. However, the plausibility of such a notion necessitates further investigation. Obviously it would also be useful to investigate if there are other techniques of measuring the usefulness of PathViz.

Apart from testing the usefulness of PathViz, we would also want to test whether the components that we used to construct PathViz function correctly in all circumstances.

In this document we focused on the notion of ontology comprehension and PathViz as an element thereof. This, however, was only a summary and the outline of a more complete work will be dealt with briefly:

- Introduction and preliminaries. This section will give a complete description of what ontologies is and the problems surrounding the understanding of ontologies.

- Literature survey. This section will focus on work that has already been done in the field. Here, we will argue that ontology comprehension is a very new field and that because we use visualisation to facilitate the process of understanding, we will focus substantially on work that has been done in the field of ontology visualisation. We end the section with a thorough overview of the work done by Bauer.

- Ontology comprehension. Here, we define ontology comprehension and we discuss why such a field is important within the realm of ontologies.

- PathViz. PathViz is the software tool that we implemented to help us solve the problems surrounding ontology understanding. Here, we will give details on the architecture and implementation of PathViz. We also discuss SuperModel and we compare SuperModel and PathViz with each other.

- Tests and experiments. We devise tests and experiments to test the usefulness/success of PathViz.

- Results and conclusions.

References

[1] Johannes Bauer. Model exploration to support understanding of ontologies. Master's thesis, Technische Universität Dresden, 2009.

[2] Johannes Bauer, Ulrike Sattler and Bijan Parsia. Explaining by example: Model exploration for ontology comprehension. 2009.