

Visualizing Concepts with Euler Diagrams

Jim Burton, Gem Stapleton, Aidan Delaney, John Howse, and Peter Chapman

University of Brighton, UK

www.ontologyengineering.org

{j.burton,g.e.stapleton,a.j.delaney,john.howse,p.b.chapman}@brighton.ac.uk

Introduction

An ontology comprises a set of statements (called axioms) that capture properties of individuals, concepts and roles. Individuals represent particular elements of the modelled domain, with concepts and roles corresponding to classes and binary relations, respectively. The primary (formal) notations for ontology modelling are symbolic, such as description logics (DLs) or OWL. To aid with accessibility and understandability, standard ontology editors often provide visualisation support. For example, Protégé¹ includes a plugin visualisation package, OWLViz, that shows derived hierarchical relationships but does not show complete information about the known relationships between the concepts.

This paper presents a new ontology visualization tool, *ConceptViz*, that uses Euler diagrams to represent concepts. These diagrams have the advantage that their topological properties reflect the semantic information that they convey; the structural correspondence between topology and semantics called *iconicity* by Peirce [3]. For instance, the containment of one curve by another reflects a subsumption relationships (i.e. set inclusion) between concepts. The asymmetric, transitive nature of curve containment reflects the asymmetric, transitive nature of (proper) subsumption. These properties motivate the choice of Euler diagrams as an effective medium for ontology visualisation. Furthermore, we combine *asserted* and *inferred* ontology information to directly visualise facts that users would otherwise need to consult several different sources to ascertain.

The ConceptViz Tool

ConceptViz has been developed as a plugin for Protégé. Figure 1 shows a screenshot from the *Pizza* ontology², which defines *Spiciness* to be equivalent to the union of three pairwise disjoint concepts – the disjointness is conveyed by the placement of the circles, while the shading tells us “nothing is spicy unless it is hot, medium or mild”. For contrast, the OWLViz view of the *Spiciness* concept, displayed in figure 2, does not show the information that *Mild*, *Medium* and *Hot* are pairwise disjoint, or that they form a partition of *Spiciness*. Several parts of the standard Protégé interface must be inspected to gather the information provided by figure 1. We see the increased co-location of information as one of the strengths of ConceptViz.

¹ <http://protege.stanford.edu>, accessed December 2013.

² Protégé OWL tutorial: <http://owl.cs.manchester.ac.uk/tutorials>, accessed December 2013.

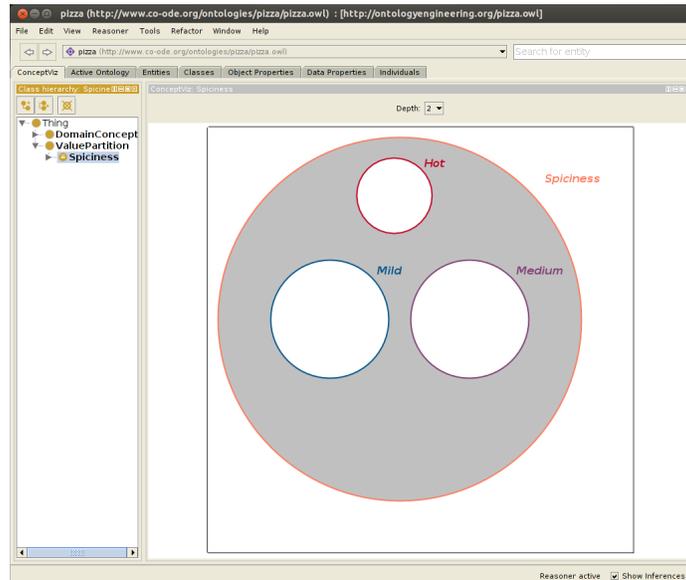


Fig. 1. ConceptViz view in Protégé.

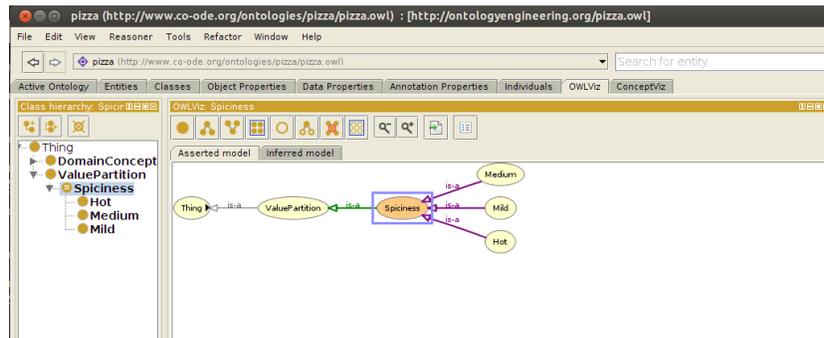


Fig. 2. OWLViz view in Protégé (whitespace cropped).

In order to produce the Euler diagram in figure 1, ConceptViz extracts information about the concepts given in the associated OWL file via the Protégé OWL API. Protégé also incorporates automated reasoners which provide the user with information that can be inferred from that which the user has directly asserted. After initiating a reasoner, Protégé users are able to inspect the inferred information, helping them to understand their ontology more fully and to identify any inconsistencies. ConceptViz interacts with reasoners to discover inferred information about equivalence, unions and so on. Figure 1, for example, includes inferred information and was produced by ConceptViz after a reasoner, started by the user, made inferred information available.

ConceptViz generates an *abstract description* of the to-be-drawn Euler diagram. The abstract description is passed to the *iCircles* library [4], which automatically produces the drawn diagram. Figure 3 shows a diagram generated by the tool alongside its abstract syntax.

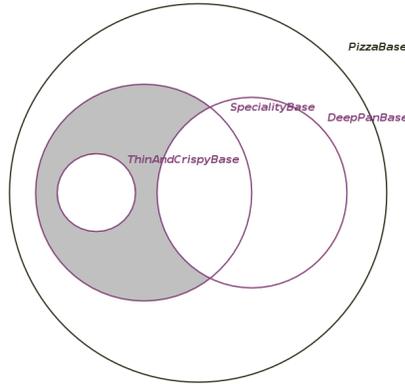


Fig. 3. A diagram generated by ConceptViz.

1. labels:
 $\{PizzaBase, SpecialityBase, ThinAndCrispyBase, DeepPanBase\}$
2. curves: $\{c_p, c_s, c_t, c_d\}$
3. regions (called zones):
 $\{\emptyset, \{c_p\}, \{c_p, c_s\}, \{c_p, c_d\}, \{c_p, c_s, c_t\}, \{c_p, c_s, c_d\}\}$
4. shaded zones: $\{\{c_p, c_s\}\}$.

Fig. 4. Abstract syntax

Vision for ConceptViz

Work is under way to extend ConceptViz to support a notation, *concept diagrams* [1], which is expressive enough to visualise entire ontologies. Concept diagrams extend Euler diagrams with syntax to represent individuals (by solid dots) and roles (by arrows); their use will allow us to provide a more compact and expressive ontology visualisation than currently exists. This work is also informed by the ontology visualisation literature. For instance, a common short-coming of existing ontology visualisation tools is the tendency to clutter [2]; concept diagrams have been designed with an emphasis on clutter reduction where possible. Finally, we intend to equip ConceptViz with editing features, enabling our goal of round-trip visual ontology engineering.

References

1. J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: A case study. In L. Aroyo and et al., editors, *The Semantic Web ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 257–272. Springer Berlin Heidelberg, 2011.
2. A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods a survey. *ACM Comput. Surv.*, 39(4):10+, November 2007.
3. C. S. Peirce. *Collected Papers*, volume 4. Harvard University Press, 1933.
4. G. Stapleton, J. Flower, P. Rodgers, and J. Howse. Automatically drawing euler diagrams with circles. *Journal of Visual Languages and Computing*, 23(3):163–193, 2012.