Semantically correct Visio

Christian Fillies[1], Frauke Weichhardt[1], and Bob Smith[2]

[1] Semtation GmbH
Potsdam, Germany
[2] Tall Tree Labs,
Huntington Beach, CA

Abstract. This paper presents some use cases of ontologies outside of the OWL community. We see these fields as initializers for non specialized users that are not familiar with the Semantic Web. Regarding these fields we experience a need for easy ontology editing, on different levels of ontology complexity. To fulfil these needs, we show how users make use of Semantic Web technology while modelling with Microsoft Visio. We explain how ontologies are used to ensure semantic consistency while flowcharting, which is the most important use case for Visio. We also present a graphical notation for authoring OWL in Visio and discuss which part of Description Logic can be expected to be used frequently.

1 SemTalk and Visio

SemTalk is a graphical editor for various modeling solutions based on Microsoft Visio. Visio was chosen as a platform because of its great graphical flexibility and its extendable design. It has a large installed base in the information worker community. SemTalk basically adds a Meta model layer to Visio, which allows specifying syntax for modeling methods on top of Visio shapes. Custom data, reporting and navigation are realized using an internal xml database.

It provides business process modeling, product modeling as well as a graphical notation for authoring and visualizing OWL. In respect to ontologies the focus of the tool is not on being a data store for large ontologies rather than an easy to use front end for manual editing of ontologies in a distributed environment of OWL aware systems. SemTalk provides consistency checking inside one Visio document and basic consistency checking between multiple models. While modeling SemTalk compares each text, which is entered by the user with a given list of ontologies.

2 Modeling with a corporate Semantic Web

Only a small section of our users create OWL models for its own sake. In most of the current models ontologies are used to normalize names of items in models made for a different purpose than authoring OWL. Examples are the process steps in a business process or the names of dimensions in a data warehouse. Another field is product modelling or knowledge management oriented models for portal building or EAI subjects. The intention of using ontologies for that is to create content which is semantically consistent with other content created in the same community. This means trying to check one model against the other semantically. People have models they use as checking or reference models. Often these models are glossaries or data dictionaries from other applications like Enterprise Resource Planning systems (ERP) or portals.

Any existing OWL or RDFS source can be used as a repository or glossary to ensure consistency. People often use lists of business objects provided by ERP vendors like SAP. For more general purposes web services like “WordNet” can be applied. In a corporate environment company or department specific ontologies are used. The resulting models are published in two ways: For end users graphical representations of the models are published on the intranet as HTML, MS Word or PowerPoint. For other modellers the model itself is available as a reusable component, e.g. a process model to be refined with subprocesses or reused as a process component. This makes all models a distributed web of knowledge.

Modelling of business processes and products in the context of a distributed web of knowledge differs significantly from the way those models have been created before. Before new terms are introduced, the user has to investigate if the term or fact already exists in the community semantic web. If the term already exists, the user model will reference that term by using the same URN and providing an URL to obtain its definition. Existing terms may be extended by subclasses or existing definitions of properties are added. If the concept is identical but the current domain requires a different name, a synonym can be added. For example a *customer* will be called *patient* in a medical domain.

The ontology contained, e.g. in a process is available for reuse in different processes in the same domain. The most common use case of ontologies in process modelling is to localize content to multiple languages. This is done by translating objects in the ontology which will automatically generate translated business processes. Sometimes ontologies created for one specific purpose can be reused for a new modelling problem in the same domain. For example a product catalogue made for the web shop can be reused in a process modelling project.

3 Ontologies for Business Processes as an example of light weight ontologies

The specification of business processes is a task executed by end users or consultants who are often specialized on process optimization or ERP systems. Those people are usually not educated in Description Logic and we do not experience a lot of enthusiasm to learn about it in order to make “better” ontologies.

For our purpose, which is ensuring consistency of other models, it is sufficient to build taxonomies, sometimes enriched with properties in order to make them more readable. Users have to learn about process modelling languages and a minimum of object-oriented thinking in order to apply the ontologies to their process models. We use subclassing, DataProperties and ObjectProperties. For process models we also add the list of valid verbs to the classes.

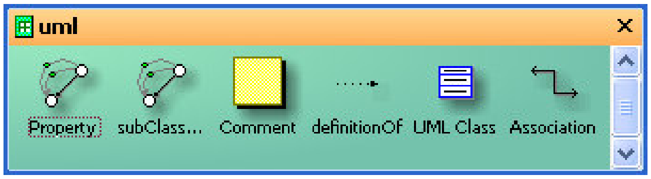

Figure 1: Class Shapes

UML-style symbols (Figure 1) are used to represent classes and connectors for "Property" and "subClassOf". There is also a specific connector named "Association" which can display cardinalities on ObjectProperties in a UML like style. DataProperties and methods (verbs) are displayed within the UML class shape. A lot of users are familiar with UML class shapes. The language which is used and supported by the SemTalk internal inference engine is similar to RDFS.

4. OWL-Ontologies as an example of heavy weight ontologies

In order to be able to express the complete language set of OWL within Visio we extended the UML shapeset with OWL specific connectors and shapes[1] (see Figure 2).

On classes we have added the constraints "disjointWith" and "equivalentClass". Different from standard SemTalk, instances are allowed in class diagrams and are allowed to be instance of multiple classes.

[1] The OWL Shapeset was jointly developed with Network Inference in order to use it as a graphical front end for their Cerebra reasoner.

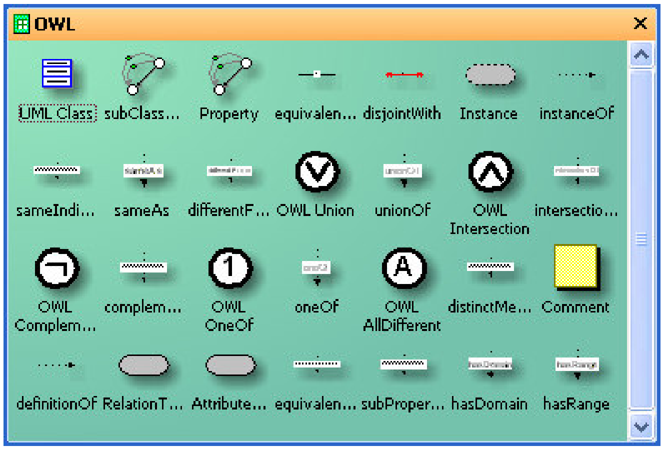

Figure 2: OWL Shapes

Anonymous classes such as *unionOf* are being expressed by a mastershape ("OWL Union") for the class and a connector ("unionOf") for the membership. Analogue combinations of master and connectors have been chosen for *intersectionOf*, *complementOf* and *oneOf.* These connectors can also be used on ordinary, named classes.

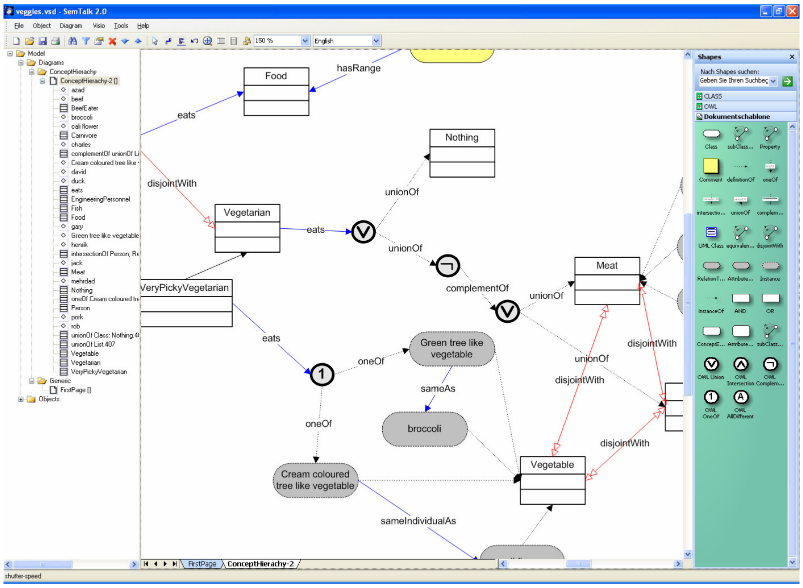

Figure 3: Anon Classes

As an addition to standard SemTalk class diagrams we have special Visio shapes to represent ObjectProperties (“RelationType”) and DataProperties (“AttributeType”) as objects in the diagram which can have graphical links “hasDomain” and “hasRange” to other objects (Figure 4).

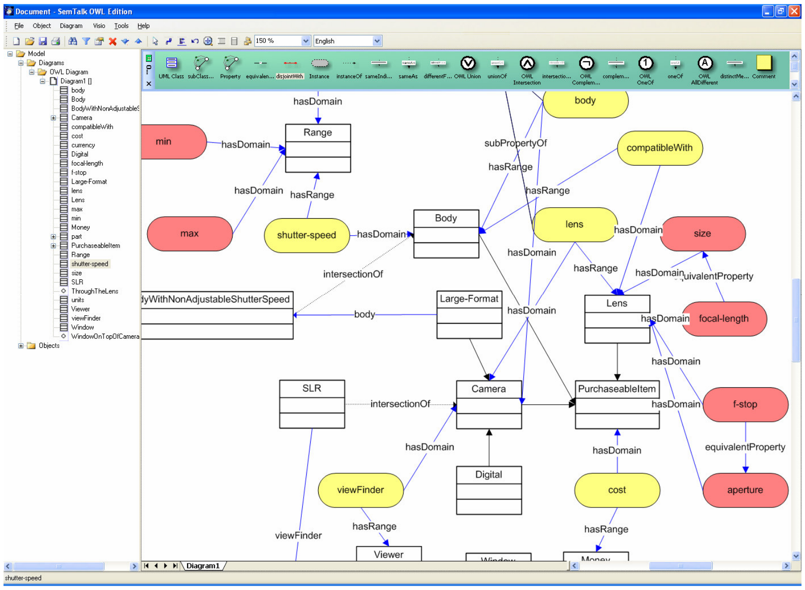

Figure 4: Properties in an OWL Diagram

Using these expressions new OWL files can be created and existing OWL files can be presented in a manually or automatically arranged way. Because predefined Visio shapes can be used to represent classes and objects, OWL models designed with SemTalk are often better understandable for non-technical end users than models created with other tools. Even if the graphical notation makes authoring OWL simpler than entering the same OWL data with other tools, it does not educate people in Description Logic. Compared to the amount of users entering knowledge using MS PowerPoint and MS Visio, the number of users specifying knowledge with OWL will be small and limited to technical experts integrating IT-Systems in EAI or Portal scenarios. We do not expect people to annotate their documents manually by modelling the contents of documents in a way inference engines can “understand” the documents. Resulting from complexity of the DL-modelling paradigm in full OWL even for stand-alone models an inference engine is needed to prove their correctness.

For some of the constraints it also makes sense to enforce consistency in a distributed environment even for taxonomies. This is especially true for disjointness, which can be violated without using any other OWL constructs other than subclassOf. A major challenge we see for inference engines is to support the distributed modelling of business processes including support for finding homonyms. Homonyms are different words having the same meaning.

5. Tools for Semantic Web Authoring

In the early nineties business process modelling has started from revolutionary ideas of Michael Hammer, who proposed business process reengineering. Pushed by the success story of ERP systems, 10 years later process modelling made its way from an academic discipline using research prototypes to a commercial component integrated in Microsoft Office used for any serious system integration.

Ontology modelling is still in its early stage. Most ontologies are made by academics using non-commercial tools which have their roots in research often in Artificial Intelligence. The “Semantic Web” in its original sense seems to be far away from reaching the critical mass required for a takeoff. But semantic technology is one of the very few technologies of the last decade which seems to ignore Gartner’s “Hype Cycle” [Eric Miller, STC05]. There has been slow but continuous growth on semantic technology and an end is not foreseeable.

Ontologies offer great value to common modelling problems especially to process modelling. Specification of procedural knowledge in processes is very common. Specification of static knowledge and rules in ontologies can be seen as an extension. Support for maintenance of static and dynamic knowledge will become part of knowledge worker’s workplace. Building end user tools for static knowledge can and will benefit from experiences made with process modelling tools.

5. Conclusion

We believe in ontology modelling as a great way of enhancing current possibilities of writing computer programs on one hand and of closing the gap between users and IT specialists on the other hand. For this we see the need to enable everybody to develop, document and maintain his or her ontology, be it in a conscious manner using DL metaphors or be it unconsciously while modelling a business process or a product. In order to fulfil this need we present a graphical way of editing ontologies, that enables all kinds of users to participate in the great vision of the semantic web.

References

[BHL01] Berners-Lee, T. Hendler, J., and Lassila, O.: published an article about the Semantic Web in Scientific American. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[GRU95] Gruber, T. (1995). Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, (43):907–928.

[HF03] van Hoof, A, Fillies, C: Das semantische Unternehmensprozessweb, Künstliche Intelligenz 4/03

[FWW02] Fillies, C., Wood-Albrecht, G., Weichhardt, F.: A Pragmatic Application of the Semantic Web Using SemTalk, WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA ACM 1-5811-449-5/02/0005

[OWL02] OWL Web Ontology Language 1.0 Reference: W3C Working Draft 29 July 2002, 12 November 2002. Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein eds. Latest version is available at http://www.w3.org/TR/owl-ref/

[STC05] Semantic Technology Conference 2005, http://www.semantic-conference.com