

Evaluation Experiment for Ontology Editors: SemTalk

The idea of the evaluation experiment for ontology editors is to model a given text with several semantic web related modelling tools. The experiment is part of OntoWeb-SIG3 EON2002 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002.

The sample text to be modelled is a natural language description of a travelling domain and the task is to model a given flight problem. This text is not a real world document from a travel agency but a textual definition of a problem space. The information given in that document can be regarded as instructions for knowledge engineers how to model this specific domain. This has a major impact on the resulting model:

- Concepts like 'Car', 'Plane' etc. usually have not to be explained to a user
- They need to be defined to do machine based reasoning in this domain
- There are a couple of ontologies out there which already describe this domain

The product idea of SemTalk is to visualize complex scenarios, often described in documents, with symbols understood by non-technical users. SemTalk is not a tool intended to be used by an high-end knowledge engineers using all features of DAML or OIL. In the demo model we have tried to demonstrate the value which SemTalk adds to a complex solution. A solution framework for reasoning should include compatible high-end editors such as OntoEdit or Protégé beneath SemTalk to express more complex logic. The focus of SemTalk is to enable domain experts to express knowledge in a way that their customers can understand it as easy and fast as possible. SemTalk is competing in the discipline of usability and not in the discipline of most sophisticated ontology modelling.

On Finding Ontologies on the Web for Referencing

In a SemTalk typical scenario we would emphasise on the statements made in one document and relate them to an external ontology. One of the most important aspects of semantic web is to make sure that people are talking about the same topic and avoid to have different representations of it. The way to do it on the semantic web is to store agreed ontologies on a common accessible place like <http://www.daml.org/ontologies> and create references to objects included in the ontologies via URN / URL. SemTalk is using the namespace of the objects for making references into RDFS / DAML. Using the namespace as the locator of an object enables us to replicate and expand objects later on.

The first step in order to create the demo model was a search on the internet for existing ontologies.

Since there is still no specific ontology search engine out there this has to be done using Google and some background knowledge. Via <http://www.daml.org/ontologies> searching for travel you will find:

http://ontobroker.semanticweb.org/ontos/com_pontos/tourism_11.daml	A couple of ontologies for travel posted by University of Karlsruhe, which are in German and can not be used for the experiment
---	---

www.daml.org/2001/06/itinerary/itinerary-ont	The interesting aspect about this one is, that the authors have been modelling “B777” and “First Class” as instances. One other reason not to use this ontology is that the current SemTalk did not understand daml:one-of and the missing ‘Restriction’ tag properly.
http://opencyc.sourceforge.net/daml/cyc-transportation.daml and http://opencyc.sourceforge.net/daml/cyc.daml	The problem with cyc-transportation ontology was, that the namespace for the objects did not match the location of the file.
http://xmlns.com/wordnet/1.6/	WordNet may be used as an RDFS Webservice in order to lookup common words and return their definition and taxonomy as RDFS.

The result of the experiment was, that we learned a lot about the syntactic variants of how DAML / RDFS has been used in existing ontologies. The SemTalk DAML import definitely has been improved. But we finally ended up using WordNet and the WordNet namespace <http://xmlns.com/wordnet/1.6/> for the classes and definitions, because it had textual definitions for those very general concepts like “Vehicle”, “Car” and “Passenger”.

Design issues for the SemTalk Model

SemTalk offers an explorer / browser to navigate the inheritance structure of the ontology. But the way SemTalk presents information to the end user is graphically.

The structure of the resulting SemTalk model in this experiment basically follows the structure of the text. We have tried to capture the contents paragraph by paragraph. For each paragraph a diagram (or “scenario”) has been built. The thumb rule for the contents of a diagram is to make not more than 7 “statements” in one drawing. The diagrams actually contain now less than 20 objects each.

Ontologies are basically are boring thing. This does not really matter as long as they are used by machines, but it is an important issue if we are using them to transfer knowledge between humans.

One way to draw attention of people to models is to use pictures and symbols. SemTalk is based on Visio with the intension to make use of the existing Visio shapes. Visio shapes can be selected from a vector graphics based library shipped with Visio, from Office Cliparts or just by using arbitrary images. For this example we found it to be the fastest and most convenient way to use images taken from Google’s image search. Using a couple of images in the graphical drawing of the ontology does not really add new information but it makes it more fun to read. Using an existing image is done by copying the jpg to the hard disk. Then drop it in the document stencil and rename it to the class name you need.

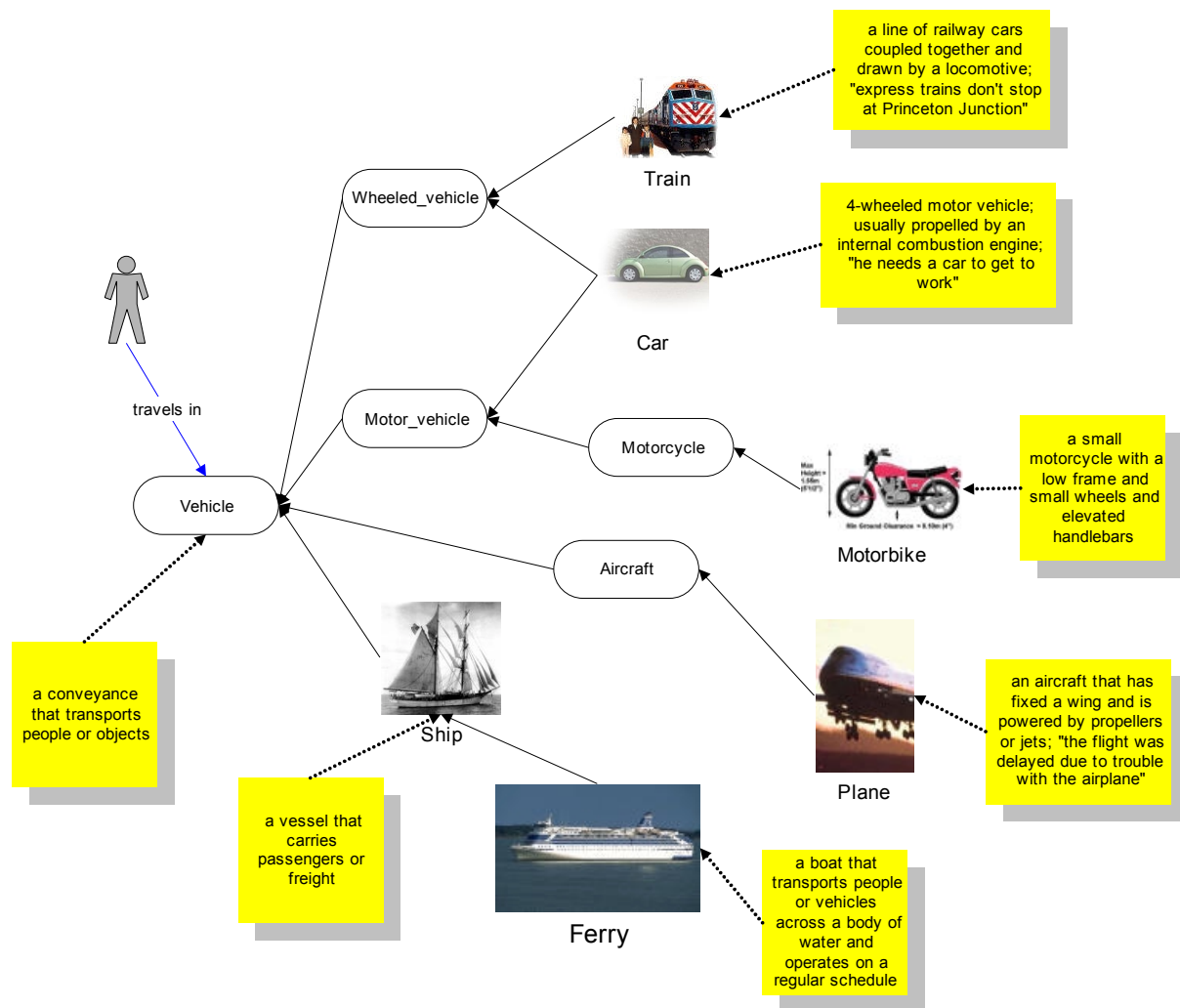


Fig.1: The Vehicle Ontology

We have attached the definition found in WordNet using a “Post-It”-style comment object. This often helps to understand the ontology even if the contents of that definition is actually ignored by any interpreter.

By assigning a Visio Symbol to a class in the ontology a kind of domain specific modelling tool for instances of the classes is created, where user can build the RDF instance model for is concrete statements using drag & drop from the ontology.

The diagrams in detail are showing:

Vehicle	A taxonomy of vehicle classes mentioned in the text
Agency	Displaying the fact stated in the text, that the agency is interested in subclasses of planes. This diagram demonstrated how to use object properties in order to express associations between objects. Since this a different statement than the vehicle taxonomy it should be visualized in a new picture instead of making the diagram to complex.
Flight	This diagram corresponds to the paragraph the text introducing attributes. The appropriate style to do this in SemTalk is to use UML-style shapes to visualize attributes. The focus of this diagram

	is to talk about the complex relations between Trip, Flight, Transportation and Topic. It also gives examples who SemTalk's inference supports property overloading (arrival, departure and used vehicle) .
Accommodation	This diagram does not add new constructs. It is basically there because it implements a lot of text about the subclasses of hotels and gives us a chance to add a picture of the tower of Chia.
Recommended Vehicle	One of the import aspects of the given text seemed to be the modelling of the relation between vehicles, transportation types and locations. This diagram shows how to do that in SemTalk again with property overloading. You may find the information that a train journey starts and end at a railway station and not at a seaport. For a train journey a train is used as a vehicle.
Destinations	This is our first instance diagram. It models the concrete destinations and continents as instances. Since we have not assigned symbols for city and continent we have used default shapes here. What we experienced as a missing feature in SemTalk was the possibility to assign individual pictures to single instances. This is currently only supported for classes.
Rules	SemTalk's native ontology modelling does not support a rule language or rule engine. Solutions like Integral, a graphical rule editor for SAP's Internet Pricing Configurator have been built on top of SemTalk.
TheTrip	This instance diagram shows simply the instances needed for John's trip.

The resulting model can be published as HTML. In the [HTML document](#) we have added source text with some hyperlinks to classes in the ontology.

We also can export RDFS or [DAML from this SemTalk model](#). Classes only, instances only or both combined in one file.

The DAML files can be

- included as markup in the original documents or
- stored as markup besides the original documents or
- published on a server as a reference ontologie or
- used as an ontology spell checker within Office XP or
-