Application of OWL 1.1 to Systems Engineering

Henson Graves[1] and Ian Horrocks[2]

[1]Lockheed Martin Aeronautics Company
Fort Worth, Texas, USA
henson.graves@lmco.com

[2]Oxford University Computing Laboratory
Parks Road, Oxford, UK
Ian.Horrocks@comlab.ox.ac.uk

Current systems engineering languages, standards,[1] and tools are restricted in certain aspects of their expressiveness and do not provide formal semantics. While there is a long history of attempts to use formal methods for engineering, up to now the relevant formalisms have generally proved too hard to use, and the tools do not scale for large and complex system development. A semantic integration framework that can integrate representations from multiple system engineering languages and tools could, however, have a significant impact on cost, schedule, and product integrity. We are exploring the potential for OWL 1.1[2] to provide such a semantic integration for the air system engineering domain. The idea is that a product development ontology can capture the meaning of concepts in a form independent of interpretation by subject matter experts, something that is not possible with current engineering languages and tools. Automated reasoning could then be used to check design properties such as consistency and conformance with specification, and the ontology could also be used to integrate information from the large number of systems used in the design of an air vehicle.

To determine the potential to use OWL 1.1 in this setting we are developing a prototypical air system ontology in OWL 1.1 and evaluating the use of the language for developing and reasoning about systems engineering concepts such as requirements and product structure. The ontology has been developed within Protégé 4.0[3] using the FaCT++ reasoner [1]. We have verified that a simplified air vehicle design specification satisfies requirements verification criteria under specific assumptions. Preliminary analysis indicates that OWL 1.1 has the potential to have a significant impact on systems engineering tools and processes. Our work to date has, however, yielded a number of interesting observations.

Our first observation is that use of foundation ontology can save a lot of effort and may even be essential to our success. In our case we are making extensive use of the DOLCE Ultra Lite (DUL) foundational ontology [2]. Using DUL has saved us a lot of time with ontology development. For example, we use and build on DUL's partonomic properties for specifying different kinds of

[1] See, e.g., http://www.sysml.org/
[2] See http://www.w3.org/2007/OWL/
[3] See http://www.co-ode.org/downloads/protege-x/

containment in the product structure; as we are not experts in mereology it is extremely useful to be able to rely on DUL's designers to have made clear and well organised distinctions in this area. We also use DUL Qualities to represent attributes and characteristics of product components, making for cleaner and more precise modelling of values such as weights, dimensions etc.

Our second observation is that OWL 1.1 works well for the representation of product structure and (static) properties. The highly expressive OWL class constructions are essential to state requirements and assertions concerning instances of a product class. However, some important features of product specifications still cannot be expressed—in particular, aggregation rules such as "the weight of a product is sum of weights of its components" cannot be expressed in OWL 1.1. Further, determining the actual values of product attributes (including simple attributes such as weight and more complex ones describing performance) is often carried out using external systems ranging from simulations to physical measurement of prototypes; in practice it would be important to integrate an OWL 1.1 ontology reasoner with such systems and to allow for information flow in both directions. Another case where integration with external systems is needed is when the parts list corresponding to an air vehicle configuration is stored in a database (as will typically be the case). It would be extremely useful to integrate the ontology and database do as to automatically create a product instance that corresponds to the conceptual model.

Our third observation is that representing dynamic properties of products, such as the change in fuel level during flight operation, presents problems for OWL 1.1. Dynamic properties are properties that change with respect to time or some operational context. DUL provides concepts needed to represent behavior, but a more expressive language (such as FOL) seems to be required in order to fully model behavioural requirements and test results relating to behaviours. We are currently exploring if/how this can be handled in OWL 1.1 as it would be very useful to have such results available in the ontology.

In spite of the above difficulties we have been successful in using FaCT++ for simple reasoning experiments to show consistency or inconsistency of product descriptions. This is very promising as existing systems engineering tools are not able to perform this kind of check. We are still experimenting with how best to divide the modelling and reasoning tasks between the TBox and the ABox: the former allows for gradual refinement and varying granularity but the latter allows for the more precise description of complex relational structures.

In summary while not a complete replacement for Systems Engineering languages and tools, OWL 1.1 could be an extremely useful component in an integrated air systems engineering environment: it is expressive enough to describe most (static) aspects of air vehicles, its formal semantics provides a precise notion of meaning, and its reasoning services can usefully augment external testing and verification. The XML syntax of OWL 1.1 may facilitate the exchange of data with database systems, but a more complete integration with other systems is still the subject of ongoing research.

References

1. Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.
2. Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with DOLCE. In *Proc. of EKAW 2002*, pages 166–181, 2002.