Integrating SysML and OWL

Henson Graves

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

Abstract. *To use OWL2 for modeling a system design one must be able to construct a Knowledge Base (KB) that can represent detailed information such as the number of occurrences of a part and interconnections between parts. SysML Block Diagrams (BD) have sufficient expressiveness to represent detailed designs. Suitably restricted SysML block diagrams can be translated into OWL2 to achieve the same result. A SysML graphical syntax can be used for the KBs which are characterized as designs. Block Diagrams have a natural model-theoretic semantic and this semantics is preserved by the translation into OWL2 for a restricted class of Block Diagrams. An implementation of a design is a parts description of the system being modeled. For a design, a KB-model will contain an implementation of the system, but may contain other entities. When additional constraints satisfied by a KB that represents a design then the KB serves as a template for the design implementations. Design KBs can be developed in engineering design tools and exported to OWL tools for analysis. This work yields a partial unification of SysML and OWL that is sufficient for modeling the structure of complex systems.*

Keywords: Conceptual Model, Description Logic, Graph-Extended DL, Ontology, OWL, Product Model, SysML, UML.

1 Introduction

Modeling is used extensively in product development to specify and analyze designs and requirements. A product model in the sense used here is a class which describes or classifies individuals in a domain of discourse. For product modeling the individuals are referred to as parts. Parts include physical parts as well as subsystems. The informal concept of a design for a system such as an automobile is that the design determines the system's possible implementations. An implementation is described by a collection of parts and interconnections between the parts. A detailed design has the additional property that it is a template for its implementations in the sense that all of the implementations have the same structure. OWL2 [13] is a natural candidate for product modeling and first steps to use OWL2 have been taken [14,4,5]. OWL reasoning tools have been used to establish consistency of a set of requirements [6]. However, to use OWL2 for modeling a system design one must be able to construct a KB that can represent detailed information such as the number of occurrences of a

Proceedings of OWL: Experiences and Directions 2009 (OWLED 2009), Rinke Hoekstra, editor.
http://www.webont.org/owled/2009

part and interconnections between parts. For example, a design for a vehicle may specify that it has exactly two identical fuel tanks with a specific connection between the two tanks.

The question of whether detailed design information can be represented within OWL2 is answered affirmatively by recognizing that SysML Block diagrams provide the expressiveness needed to represent detailed design information [1] and by recognizing that Block Diagrams can be translated into OWL2. OWL2 KBs can be constructed to represent detailed design information such as part occurrences and connectivity. SysML [12] is a language designed specifically to represent structure and behavior of systems for product development and engineering analysis. It is an OMG Standard and has well developed authoring tools [3, 2]. SysML provides a graphical syntax which is particularly useful for human comprehension; however SysML lacks a formal semantics. While OWL and SysML use different names, both have terminology for individuals, classes, and properties. SysML has terminology for operations which is lacking in OWL2 and OWL2 has term constructions for classes that SysML does not have. Both have data types and properties can have data type values.

The translation of a restricted form of SysML Block Diagram provides a semantics for the subset of SysML constructions translated into OWL2. The question of whether the SysML to OWL2 translation preserved the intended SysML semantics, led to the question of finding a formal semantics for SysML. This question can also be answered affirmatively. A SysML Block Diagram is a kind of first order equational logic [10]. The equations express syntactic relationships in the diagram. A Block Diagram provides an abstract syntax for the kind of terms used in constructing Block Diagrams. Logical axioms are expressed using equality, instance of, and subclass relations between terms. Block Diagram specific structure can be expressed using equality, instance of, and subclass relations The informal interpretation of a BD can be captured by a definition of a BD-interpretation. BD-interpretations can be used to define a formal model-theoretic semantics. A BD-theory can be defined in the usual way as sentences that are true in all of the models. However, logical axioms can be constructed which can be used to characterize the BD-theories. The discussion here will be limited to the SysML Block Diagram constructions that can be directly translated to OWL2. The more general form of Block Diagram formalization and its semantics will be treated elsewhere.

The KB representing the design will contain the class that represents the system and will have classes for the other parts that make up the system. A KB suitable for representing designs will have distinct “hasPart” properties with domain and range classes that represent the graph structure of the BD. Each class that has immediate parts has a distinct hasPart property for each class of parts. Cardinality restrictions on these properties determine the number of instances of the class that occur in an implementation. An implementation of a design is a parts expansion of the class representing system being modeled rather than being a model of the entire KB. A model of a design KB may contain other elements and an implementation of the design may contain extraneous parts. No cardinality restrictions on the classes are imposed. This approach to defining design implementations is different from the Description Graph extension [11] which restricts the models to rule out the unintended models.

2 Block Diagram Semantics

Figure 1 illustrates a simplified design representation for part of a vehicle as a Block Diagram. The diagram consists of blocks (classes) which are connected to parts by connection associations using SysML terminology. Multiple kinds of connections occur in product models. Connections commonly employed include fuel, electric, and hydraulic flow, as well as physical connections. In general a Block specializes a general part class; block attributes (properties) are used to classify different kinds of parts. A complete treatment would subclass blocks from a general Part class. For this discussion, Block Diagrams are restricted to have only Associations with no operations.

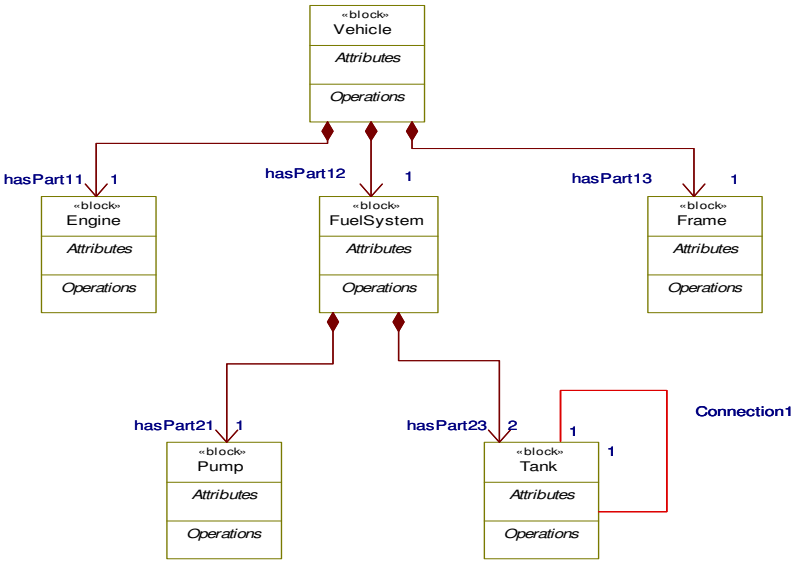

Fig. 1. The SysML Block Diagram represents an incomplete design for a vehicle. Each association in the diagram has a unique source and target class. Vehicle is a root class under the hasParts associations. The diagram specifies that FuelSystem has exactly two tanks and that they are connected with Connection1 association.

The diagram in **Fig. 1**contains a distinct named association for each class in the diagram. There are no associations without a Source and Target class. The named hasPart associations however, can be viewed as sub-associations of a general hasPart Association. The diagram has a graph structure labeled by the two kinds of associations. In general, a BD will contain subclass relations and instances. However, the Vehicle System Block Diagram does not contain either. The blocks with the part associations are a tree within the Vehicle System BD using the hasPart properties. Expanding the hasParts associations provides a vehicle parts decomposition with connections between parts. The Vehicle block in **Fig. 1** may have many instances, which represents the fact that there may be many air vehicle instances. Similarly, the other classes in the diagram may have many instances.

However, the parts and connection structure are defined by the diagram. Cardinality restrictions on the associations are used to specify how many parts an instance of a class has.

2.1 Abstract Block Diagrams

The Vehicle System Block Diagram can be generalized to produce an abstract definition. An OWL Block Diagram signature has three kinds of sorts; *Individual*, *Class*, and *Properties*. The Block Diagram operation symbols are in the table below. A Block Diagram Signature is finitary in that each of the sorts contains a finite number of symbols. Let *Term* be the union of *Individual*, *Class*, and *Property*. The usual notation for the BD-operation application will be used which is infix in the case of equality and instance. The symbol N will be used as a constant for numbers.

Dom : *Property* → *Class*,	Assigns a domain class to a property
Range: *Property* → *Class*,	Assigns a range class to a property
Restrict : *Property* → N	Restricts the number instances of the domain class
* : *Property* → *Property*	The inverse property
o < *Property* , *Property* > → *Property*	Composition of properties
Class, subclass *Class*	
Property subrelation *Property*	
Term = *Term*	Equality of terms
Individual : *Class*	instance of class
< *Individual*, *Individual*> : *Property*	instance of property

An OWL BD is a special case of a more general concept of BD whose signature has operator symbols corresponding to SysML Operators. Individuals are a special case of operators which have no arguments.

An abstract Block Diagram is a finitary collection of symbols of the three kinds, *Individual*, *Class*, and *Properties* , together with equations, subclass, and membership relations which directly express the syntactic information contained in the graphical syntax of a Block Diagram. The Vehicle System BD is an abstract Block Diagram with the following signature:

N	- number type
Vehicle, Frame, Engine, FuelSystem, Pump, Tank	- class symbols
hasPart11, hasPart12, hasPart13, hasPart21, hasPart22	- “hasPart” properties

Connection1	- property used as an example of connection associations between classes

To define the Vehicle System Block Diagram, we use equations which specify the domains, ranges, and cardinality restrictions of the diagram. For example:

Dom(hasPart11) = Vehicle,
Range(hasPart11) = Engine
Restrict(hasPart11) = 1.
Dom(Connection1) = Tank
Range(Connection1) = Tank
Restrict(Connection1) = 1
Restrict(Connection1*) = 1

The Vehicle system Block Diagram does not have any subclass relations and does not have any individual declarations. Block Diagrams may have subclass relations and individual declarations. The reason that these are not included is that this BD is a detailed design. The informal semantics for the collection of "hasPart" properties are that for each property in hasPart, the domain and range classes are distinct and span $\mathscr{Class}$.

2.2 Block Diagram Semantics

A model-theoretic semantics can be defined in terms of interpretations of the BD signature within a domain of individuals. The meaning of a Block Diagram can be expressed in terms of interpretations of the diagram within a domain if individuals. This semantics captures informal meaning and is also consistent with an OWL2 KB semantics. For a Domain $\mathscr{D}$, a BD interpretation in $\mathscr{D}$ is a mapping

^ : Block Diagram Signature → Sets($\mathscr{D}$)

which maps the class symbols to sets and properties to binary relations defined for the domain and range classes, and which preserve any subclass and subproperty restrictions. The interpretation of the restriction operation corresponds to the OWL class construction "R exactly k A" where R is a property and A is a class and in the restriction interpretation of a BD property R, A corresponds to the Range(R). For sets A and B the notation <A, B> is used for the Cartesian product of A and B. The notation

R: A→ B

is used as an abbreviation for

Dom(R) = A and Range(R) = B.

A^ is a subset of $\mathscr{D}$
R^ is a subset of the product < $\mathscr{D}$, $\mathscr{D}$ >
If R : A → B, then R^ is a subset of the product <A,B>
If R : A → B, then R*^ = R^*

If Restrict(R) = k, then Dom(R) is the set R^ exactly k Range(R^)
If A SubClass B then A^ is a subset of B^

A theory for a specific Block Diagram may be defined as the sentences which are true in any interpretation of the Block Diagram. Logical axioms can be specified by formulas that use the instance, subclass, and equality expressions. Examples of the axioms are shown in the table below. The left and right columns are logical equivalence. A class constant Thing and a property constant ID are added to a BD signature.

Thing	For any a. a : Thing
C SubClass D	For any a. a: C implies a:D
Dom (R) = A	For any a, b . <a,b>:R implies a: A
Range (R) = B	For any a ,b . <a,b>:R implies b: B
R : A → B	R* : B → A
R*(R) = R(R*) = ID	

A BD theory preserves the logical axioms.

2.3 Block Diagram Representation in OWL2

An OWL2 KB is described by a signature of *Individual, Class,* and *Property* with the OWL term constructions and axioms for equality, subclass, and instance, as well as the term constructions. Examples of these axioms for existential and universal restriction are:

Class Term	**Axiom**
R some A	For any a (there exist b (<a ,b> : R and b : A))
R any A	For any a (for any b (<a ,b> : R implies b : A))

This description of OWL2 is also presented as a partial equational theory with a richer set of class term constructions than is available in a BD theory. An OWL2 KB may be generated from a BD theory adding the OWL class term constructions. Each BD restriction axiom of the form "Restrict(R) = k" where R in *Property* translates into an OWL2 axiom of the form "Dom(R) subclass R exactly k Range(R). The semantics is preserved by the translation of a Block Diagram into a KB. The properties defined for a BD-interpretation are preserved by the OWL2 models.

2.4 Design Block Diagrams

The Vehicle System BD has a tree structure using the hasPart properties with Vehicle as the root. More generally, a Design BD is a BD which has a subsort *hasPart* for which

< *Class*, *hasPart*, Dom, Range >

where the domain and range equations define a tree on classes from *Class* for each property in *hasPart*. For an interpretation, the domains and ranges of the properties in *hasPart* are a disjoint covering of the union of the carriers for the class symbols. A

Detailed Design BD is a BD for which each class A is equal to a conjunction of its has property restrictions, i.e.,

A = R1 exactly k1 … and Rn exactly kn

where each Ri is a member of *hasPart*, dom(Ri) = A, and ki is an integer.

An instance of the top class determines a recursively constructed list of parts and connections by expanding the root class definition. Properties of instances of the class of systems under design can be proved using properties of the design realization. A Design KB can be defined as a KB whose signature contains a BD. All of the implementations of a Detailed Design KB have the same structure.

3 Conclusions

This work is intended as a step toward enabling product developers to use SysML tools with their graphical syntax and export the result to reasoning tools for design analysis and verification. A partial integration is achieved by translating a restricted SysML Block Diagram syntax into an OWL2 KB. The models of the OWL2 translation of a BD are the same as given by the BD interpretation semantics. Thus the translation preserves the BD intended semantics. The result provides an OWL2 semantics for a portion of SysML. Conversely, SysML graphical syntax to be used within OWL2 for the design KBs. A restricted set of Block Diagram constructions that are needed in representing product designs can be used to generate a KB which preserves the structure of the block diagram. The Block Diagram interpretations for the restricted Block Diagrams preserve the structure that a model of an OWL2 KB preserves and so this logic extends OWL2.

The Block Diagram to OWL2 translation provides a method to construct an OWL2 KB that can represent structural design information such as parts decomposition and connectivity structure. The translation indicates that OWL 2 is sufficient for representing product structure. The capability of OWL2 to represent parts and connectivity structure depends on the KB having a root class which has a parts decomposition graph in any KB model. For Design KBs the graphical syntax of SysML can be used to develop OWL KBs. This solution does not depend on restricting the models as is done in the Description Graph extension to OWL2.

Of course to realize the full goal using formal reasoning tools in product development would require a formal semantics for a much richer subset of SysML, for example including ports with their interfaces and SysML operations. A general Block Diagram theory can be presented as a partial membership algebra signature. Based on the understanding of the interpretation semantics for Block Diagram, one can define a Block Diagram theory as a Horn Class theory containing logical axioms and prove that the BD theory models are exactly the BD-interpretations. A BD theory provides a formal semantics for the constructions covered by the BD signature. A BD theory is more general than an OWL2 KB as it contains operators. These results will be addressed in a following paper.

References

1. Bock, C., *UML Composition Model*, Journal of Object Technology, vol. 3, no. 10, November-December 2004, pp 47-73.
2. Friedenthal, S., Moore, A., and Steiner. F., OMG Systems Modeling Language (OMG SysML™) Tutorial, INCOSE Intl. Symp, 2006.
3. Graves, H., Guest, S., Vermette, J., and Bijan, Y., *Air Vehicle Model-Based Design and Simulation Pilot*, Spring 2009 Simulation Interoperability Workshop (SIW)
4. Graves, H. *Design refinement and requirements satisfaction*, Proceedings of 9th NASA-ESA Workshop on Product Data Exchange, 2007.
5. Graves, H., *Ontology engineering for product development*, Proceedings of the Third OWL Experiences and Directions Workshop, 2007. Available at www.webont.org/owled/2007/PapersPDF/submission_3.pdf
6. Graves, H., Horrocks, I., *Application of OWL 1.1 to Systems Engineering*, OWL Experiences and Directions April Workshop, 2008.
7. Graves, H., *Representing Product Designs Using a Description Graph Extension to OWL 2*. OWL Experiences and Directions October Workshop, 2008.
8. Graves, H., Leal, D., *Using OWL 2 For Product Modeling*, Proceedings of 11th NASA-ESA Workshop on Product Data Exchange, 2009.
9. Haley, T., Friedenthal, S., *Assessing the application of SysML to systems of systems simulations*, Proceedings of the Spring Simulation Interoperability Workshop, September, 2008.
10. Meseguer, J. Membership Equational Logic as a Logical Framework for Equational Specification. Proc. 12th WADT Workshop on Algebraic Development Techniques, Springer LNCS, 1998.
11. Motik, B., Grau, B., Horrocks, I., Sattler, U., *Representing Structured Objects using Description Graphs*, AAAI, 2008.
12. OMG Systems Modeling Language (OMG SysML™), V1.1, November 2008.
13. OWL 2 Web Ontology Language, W3C Working Draft 11 June 2009.
14. Price, D., Bodington, R., *OWL, description logic and systems requirements satisfaction*, Proceedings of 8th NASA-ESA Workshop on Product Data Exchange, 2006.