

INsight to Diverse Information using Graphs and Ontologies

Collaborative MBE across Diverse System Models Including AADL

steve.vestal@galois.com ed.sandberg@galois.com

Agenda

- Example Open Model-Based Engineering Environment (OpenMBEE)
- Characteristics common to digital models
- INsight to Diverse Information using Graphs and Ontologies (INDIGO)
- Example use cases
- Interoperability features and guidelines



Open Model-Based Engineering Environment



OpenMBEE is an open source collaborative engineering system. It enables engineers to work in the language of their choice and easily share and document their work across other tools.

- NASA initiated
- NumFOCUS foundation open source project
- <u>https://www.openmbee.org</u>
- <u>https://groups.google.com/g/openmbee</u>

See also

- OSLC <u>https://open-services.net/</u>
- SysML-v2 <u>https://groups.google.com/g/sysmlforum</u>
- Open AADL http://www.openaadl.org/





Example Open MBEE



OSATE

- Move diverse models in and out of a FlexoMMS server
- Explore configurations of models in INDIGO
- Identify API alignments, gaps, standards-based solutions



Example Modeling Languages & Tools Features What is a Digital Model?

• Modelica

- Specification (4 "model" instances): can be translated into a simulation model, other tool usages allowed but not specified
- In practice: concrete textual syntax, translate to a dynamical systems simulation model
- Semantics: systems of differential equations
- AADL
 - Specification (211 instances): model describes a system as a hierarchy of components with their interfaces and interconnections
 - In practice: concrete textual syntax, set of <u>package</u> files required to <u>instantiate</u> a <u>system implementation</u>
 - Semantics: state machines, scheduling theory, temporal logics, assume/guarantee, fault trees, Markov chains,...

• SysML-V2

- Specification (486 instances): represented according to the syntactic requirements (abstract and concrete syntax)
- In practice: a SysML modeling tool is conformant with either the abstract or concreate syntax and creates, manages, analyzes, visualizes, executes or performs services on SysML models
- Semantics (KerML): model-theoretic

• Resource Description Framework (RDF)

- Specification (model=graph): RDF graphs are independent of syntax. Graphs not RDF-isomorphic are distinct graphs.
- In practice: sets of triples <resourceIRI> <propertyIRI> <resourceIRI or dataValue>, multiple syntaxes, flexible and varied
- Semantics: referents and entailment regimes, flexible and varied

• Web Ontology Language (OWL)

- Specification (model=ontology): defined by behavior and structure, multiple standard and common syntaxes, content identified by self-declared ontology and version IRIs
- In practice: RDF-based syntaxes, syntactic/shape validation, logical consistency validation
- Semantics: model-theoretic, decidable description logic, entailment regimes

Characteristics of Digital Engineering Models

Digital Model: a body of information that

- Evolves over time
- Can be expressed in one or more multi-media syntaxes
- Has modularization and dependency features
- Copies can be stored in multiple storage systems
- Has a principle of individuation (is distinct from other models)
- Has zero or more formal (math-based) semantics
- Has one or more tools, vendors, user interfaces
- Has information that relates to information in other models
- Often analyzed to provide parameters for other models
- Often overlap with other models

galois

INDIGO

galois

INsight to Diverse Information using Graphs and Ontologies



Populate Library of Models and Ontologies



Copyright Galois, Inc. 2024

Choice of model File Edit Navigate viewer can be opened 📑 🗕 🖪 👘 📮 🕻 Project Explorer on a set of models. > > b cubesat_trades iresat > firesat_trades AADL domain ன (R) AADL L > Section FireSat Trace ontology (terms 🚧 http://www from SAE AS5506) >>> http://www 🦟 (R) http://w SysML Model firesat_tradespace.1 > in add: http://w ✓ ★ CSAH_examp AADL Model > [] aadl:access I] aadl:com 😂 Indivic 🗸 🚞 Proper Users interactively ⊃ aac query-with-reasoning ⊃ aac ⊃ aac by selecting classes, \supset aad

⇒ aadi

properties, etc.

Explore Set of Models (tree viewer)

FireSat is a SysML satellite mission specification. CSAH is an AADL cubesat design & analysis architecture.

e Search Project Run Window Help	
▙ ▾ ! ▾ ! 뭘 ▾ ?! ▾ ♡ ઃ ♡ ઃ ◇ ▾ ♂	Q i 🖻 🕹 💁
Ontology Manager × 8 □ □	<pre> p firesat_tradespace.1 tree × </pre> 8 □ □
space pace anguage Terms deSpace combined tree view v.sanford.friedenthal.net/firesat v.your.name.here/CSAH_example vww.your.name.here/CubeSat_Design_Space_O292	 Communication ->sysml:contains->csds:CSAH_example_Instance.CubeSat.Comm ->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.Comm.antenna ->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.Comm.telemBus ->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.Comm.transceiver ->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.Comm.transceiver ->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.Comm.transceiver ->sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4 ->sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4 ->sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4 ->sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4 >sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4
operties tree ww.adventium.com/aadl le: http://www.your.name.here/CSAH_example sConnection	 :IBD_Subsystem :Power ->sysml:contains->CubeSat_Design_Space_O292:PowerSystem ->sysml:contains->firesat:Data.Spacecraft Mission Context.6-Spacecraft.Physical Architecture.4 ->sysml:contains->csds:CSAH_example_Instance.CubeSat.powerSystem ->aad_sasSubcomponent->csds:CSAH_example_Instance.CubeSat.powerSystem.powerBus
Select class Deselect class members Select class and direct subclasses Copy Select as a root in model tree viewer Select all members as roots in model tree viewer	->aadl:hasSubcomponent->csds:CSAH_example_Instance.CubeSat.powerSystem.solarPanel ->aadl:hasSub_mponent->csds:CSAH_example_Instance.CubeSat.powerSystem.solarPanel ->aadl:hasSub_mponent->csds:CSAH_example_Instance.CubeSat.powerSystem.battery ->aadl:hasSubcom_onent->cs ->aadl:hasSubcom_onent-

following user-selected

properties, classes, etc.

Different sets can be opened using different kinds of model viewers at the same time.

User selections can be saved as a viewpoint ontology, a set of rules to reconstruct the view.

API allows other viewers to be added as plugins.

Explore Set of Models (graph viewer)



FHOWL Literate Ontology

😻 SinglePointFailurePattern.fwl 🛛 💐 class_examples.fwl 🛛 💐 property_examples.fwl 🗡

Example Property Definitions

https://camet-library.com/camet

Here are examples of property definitions. A property defines a relation between individuals (elements) of a model, analogous to the way a class defines a set of individuals. The members of a property are called property instances (a.k.a. triples, statements). This ontology references (imports) the class examples ontology.

propA (object property): primal

A primal property is one that cannot be defined using other terms in this or other referenced (imported) formal ontologies. These are axiomatic classes whose semantics must be known or explained to the user by other means. References to other information such as standards, textbooks, or documents in other formats should be cited.

A primal property definition must have a propertyType, one of "object" (property goes to an individual), "datatype" (property has a datatype value), or "annotation" (OWL annotation property).

propData (datatype property): primal

The datatype value is one of the XML Schema built-in datatypes plus "anyType".

propD (object property): primal

Another primal property.

inversePropA (object property): inverse propA

An inverse can only be declared when the **propertyType** is "object". A literal (e.g., a mathematical value) cannot have as a property an element of a model.

🖩 Design 🗎 Source Preview

Literate format interleaves formal ontology definitions into an XHTML document.

FHOWL extends OWL with structured classes and properties defined by patterns that recognize subgraphs in a given context.

4/3/2024

galois Ontologies for Multiple Domains and Sources

Domain ontologies are widely used to provide a vocabulary with definitions. They serve as schemas for RDF graphs that support logic-based reasoning.



Interoperability Features and Guidelines

Resource Description Framework (RDF) exchange formats

- Graph structure vs folders of files of lines of text
- Semantic referents and entailment regimes
- SPARQL standard query language and protocols
- OWL model-theoretic semantics and reasoning tools
- Aligned with other web standards, XML, etc.
- Widely used and supported, e.g., OSLC, WikiData, SNOMED, PubChemRDF

Domain Ontologies

- Vocabulary of terms with definitions relating them
- Formal OWL ontologies
- Multiple types
 - Schemas for languages, e.g., terms from SAE AS5506
 - Translations, e.g., aadl:bus defined for 3D CAD patterns
 - Viewpoints, e.g., selections and rules to generate diagrams

alois

Interoperability Features and Guidelines

Open source conversion frameworks, e.g.,

- Eclipse Lyo (OSLC)
- Ecore2OWL
- ReDeFer (XSD2OWL and XML2RDF)
- STP2OWL

Standards and guidelines

- Ontology/namespace/base IRIs to uniquely identify models
- Version IRIs and properties
- Modeling language domain ontologies
- Standard domain ontologies, e.g., OSLC, QUDT,...

galois