Ontology Engineering for Systems Engineering

techinvestlab.ru

Ontology Summit
26-jan-2012

TechInvestLab ontology-related projects

- .15926 – software platform for ISO 15926 (currently available freeware .15926 Editor version 0.86)
- Praxos – development of engineering enterprise (i.e. enabling system) ontology-based methodology

We have about 100 members in Russian-speaking community around this two projects.

We maintain communications with INCOSE and have regular workshops that deals with usage of ontology in systems engineering.

Also we have participate in industry-wide life cycle data integration projects in nuclear power plant and shipbuilding industries of Russia.

Formalization of Engineering

Languages of systems engineering and specialty engineering:

- Informal (text and pictures)
- Semiformal (diagrams and drawings: “pseudocode”)
- Formal (computer interpretable -- rigorous --domain-specific languages)

More formal languages → less errors in system definition → less errors in system realization

Ontology engineering helps advance formalization of systems engineering domain-specific languages that was not error prone before:

- Requirements
- Architecture
- Assurance cases

This is not cases for traditional engineering mathematical (differential calculi) languages.

The case is to provide language workbench for DSLs of complex engineering project (like language workbench for software DSLs).

Engineering Method formalization

1st step: ontology-based system definition → early engineering collision discovering and prevention

2nd step: ontology-based engineering method definition → generative design *and* generative manufacturing

Need: method description formal languages, thus ***ontology of a method***

Counterintuitive (not folk) ontologies

- Contemporary ontologies “folk”/“common sense” by definition (“shareable understanding”, intuitive, understandable by common public)
- Contemporary engineering often counterintuitive, not “folk” or “common sense”, most advanced engineering methods understandable by few talents. Engineering State-of-the-art will be “common sense” after decades.

Need for counter-intuitive ontologies (like ISO 15926, IDEAS, HQDM). True ontology, not linguistics/terminology (knowledge about things, not about descriptions of things).

* * *

Still need for “common sense” and “folk ontology” (many engineering standards based on not-state-of-the-art engineering). [Example of CYC with medical reports]

Immediate usage: PLM data models

Problems in definition of ontology for system-of-interest (PLM repository):

- ontology of modularity (system with subsystems, interfaces and catalogs)
- Geometry (shapes and mereotopology)
- Causality, mapping, “meta”, requirements, etc. traces relations and modalities
- Design/solution variants and risk modeling
- Socio-technical systems representation
- Real-time modeling (multi-physics). Example: Simantics that not too similar to ordinary PLM
- Multiparadigm Software and computers (cyber-physical systems)

Problems in definition of ontology for enabling system (PLM workflow/issue tracking – ONTOLOGY OF SYSTEM ENGINEERING as a discipline):

- System life cycle models, adaptive case and issue management, situational method engineering, process and project management: engineering activity ontology
- Motivational models for human action in enabling systems/systems of systems/extended organizations/eco-systems (praxeology)

Hint: there is consensus about «what is system», but still no consensus about «what is systems engineering»...

Legacy standards, regulations, prototypes

Legacy standards, regulation and prototype designs and projects are mix of natural language sentences, semiformal and formal (often graphic) languages.

We need new type of NLP: *engineering artifact* (diagrams, drawings, texts with tables and formulae) *processing.*

This needs combined usage of terminology/semiotics and ontology.

Ontology evolution

“Eternal classes” stable ontologies is not convenient for concurrent ontology engineering (with several teams that work in ontology discovering mode and perform engineering project) in first-in-a-kind engineering projects (that are very often in systems engineering). We need:

- less formal semantics, more formal pragmatics.
- Multi-agent belief revision theory
- Separation of administrative and ontology domains (units of ontology maintaining/editing/communication/library granularity and units of belief revision)

Engineering method and logic

- Current ontology engineering is based almost entirely on classical logic
- Systems engineering is based on engineering method that is usage of heuristics (non-traditional logic).

We need align engineering “logic of heuristics” and current logic of ontology engineering.

May be we need to go beyond “logic” (e.g. use category theory for ontology representation and ontology-based computations).

Computation with ontologies: induction, deduction and abduction

- We need not knowledge capture and preserving but “knowledge computations”
- Current ontology engineering tell not much about computational models for ontologies (exept separate “reasoners” for ontology repositories).
- We need not only reasoners with support of induction and deduction, but capable of abduction (discovering and usage of analogy). Thus we should start discussion about cognitive architectures.

Engineering megamodels(*) is HUGE and DIVERSIFIED. This is not similar to other types of “big data”.

Engineering megamodels require new computation architecture with correspondent software and hardware architecture.

(*) Megamodel -- design and project models along with pertinent reference data, meta-models and other needed modeling artifacts. Term “megamodel” suggested by AtlanMod group.

Architecture of ontology-based engineering information system

Contemporary enterprise architecture are based on Enterprise Service Bus.

We need Systems Engineering Service Smart Bus that:

- Ontology-based (access to reference data)
- Have interfaces as SOA for data services (with intellectual knowledge discovery)
- Have defined transport layer (knowledge routing)
- Have defined representation converting (adaptors) for knowledge editing and processing applications
- Support not only knowledge about system-of-interest but also knowledge about enabling systems (organization, methodolody etc.)

This architecture is aimed for PLM federation of the future.

Questions?

Anatoly Levenchuk,
ailev@asmp.msk.su

Victor Agroskin,
vic5784@gmail.com

TechInvestLab.ru
+7 (495) 748-53-88