Observing Formal Ontology

FROM THE STREET

A Personal Odyssey in Computerized Representations

Steve Ray
Carnegie Mellon University
FOIS 2012
Senator Jay Rockefeller discusses the factory of the future with NBS Researchers - May 4, 1987
to model...

Directed Graph

NIAM
... incorporated into a Control Architecture
... introduced to a standards committee
ISO 10303-49 Process structure and properties

Recommendation on U.S. TAG vote on Part 49.

Steven R. Ray
NIST

I recommend that we vote NO on the circulation of the draft as a DIS, for the following reasons:

Major Comments against Part 49.

1. There is no capability within the model to support the representation or binding of variables within a plan. This has a number of serious consequences:

   a) On page 1, it is stated that the "specification of alternative process plan definitions" is within the scope of the Part. Without the use of variables, it becomes impossible to formally characterize conditions under which different alternatives should be executed.

   b) It becomes impossible to represent derived relationships among processing parameters using equations or algorithms.

   c) Any parametric characterization of information within a process plan cannot be supported.

   The current model, the only way to approach the required textual descriptions of conditions (see the earlier symptom that an important aspect is arithmetic). Please note
STEP Integration of Parts

1993 Version of Part 49

Abstraction & “Cleansing”

1995 Version of Part 49

“A process has some sort of relation to another process”

Resources

ARM of AP 1

ARM of AP 2

ARM of AP 3

MTs & AIM of AP 1

MTs & AIM of AP 2

MTs & AIM of AP 3

Resource Model 1

Resource Model 2

Resource Model 3

Resource Model 4
Start Over – Do it Right
Unified Process Specification Language: Requirements for Modeling Process

Craig Schlenoff
Amy Knuttila
Steven Ray

Requirements Categorization

<table>
<thead>
<tr>
<th></th>
<th>Representational Requirements</th>
<th>Functional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>e.g. resource, task</td>
<td>e.g. extensibility</td>
</tr>
<tr>
<td>Outer Core</td>
<td>e.g. conditional task</td>
<td>e.g. exception handling</td>
</tr>
<tr>
<td>Extensions</td>
<td>e.g. process performance measurements</td>
<td>e.g. resource monitoring and feedback</td>
</tr>
<tr>
<td></td>
<td>(Analysis)</td>
<td>(Analysis)</td>
</tr>
<tr>
<td>Application-Specific</td>
<td>e.g. non-machining times (Production</td>
<td>e.g. dynamic rescheduling (Production</td>
</tr>
<tr>
<td></td>
<td>Scheduling)</td>
<td></td>
</tr>
</tbody>
</table>
Appendix A: Requirement Groups vs. Representational Constructs

(Requirements in each group are listed on the following page)

<table>
<thead>
<tr>
<th>Requirement Groups</th>
<th>Resource Representation and Characteristics</th>
<th>Plan/Task Representation</th>
<th>Resource/Task Characteristics</th>
<th>Precedence/Sequences</th>
<th>Constraints</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object-Based</td>
<td>Xxxxxxxxxxx</td>
<td>Xxxxxxxxxxx</td>
<td>Xxxxxxxxxxx</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Constraints-based</td>
<td>Xxxxxxxxxxx</td>
<td>Xxxxxxxxxxx</td>
<td>Xxxxxxxxxxx</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Graph-based</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Relationships</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Conditions</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Inference</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
<td>XXXXXXXXX</td>
</tr>
<tr>
<td>Task Decomposition</td>
<td>-</td>
<td>Xxxxxxxxxxx</td>
<td>Xxxxxxxxxxx</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Annotations</td>
<td>-</td>
<td>Xxxxxxxxxxx</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Abstract

Process Specification Language (PSL) project is to investigate a set of process information to enable sharing of process data among applications. This paper focuses on the second phase of the project to determine how well existing process representation methods for specifying processes are in Phase One. This analysis will be used to develop a comprehensive language and will promote the level...
Proceedings of the Process Specification Language (PSL) Roundtable

Craig Schlenoff, Amy Kantilla, Steven Ray

Abstract

In April, 1997, the Process Specification Language (PSL) Project held a Roundtable discussion at the National Institute of Standards and Technology (NIST). The goals of the Roundtable were to assemble key champions and stakeholders of various approaches towards process representation in order to discuss the relative merits to reach consensus on a language architecture and to establish a technical approach for proceeding. It was agreed that the language architecture should be based upon a formal semantic foundation, upon which would be layered a number of syntactic mappings, each with one or more presentations.

In discussions about principal concepts of any process representation, it was agreed that “process” and “participant (resource)” are basic. A number of possible other concepts were suggested, but no consensus was reached. Additionally, five potential uses for the PSL were identified and discussed. They were: 1) provide a description of a process that has already occurred; 2) provide a “recipe” (prescription) describing how a process can occur; 3) provide a semantic model to determine concepts and establish the scope of enterprise systems, enable interoperability between manufacturing systems, enterprise systems, and consumers; 4) enable technology transfer between manufacturing and other dis...
“If you really want to do it right, use formal logic”

- Somebody at the workshop April 1997
  (Gruninger? Menzel?)
Tactic

1. Create it more or less completely

THEN

2. Create a standards committee
PSL – Process Specification Language

- ISO 18629
- 2004-2006, All Parts of the International Standard Approved
- Model theoretic
- Fully axiomatized
- Based on invariants
- Supported by proofs
From syntax to semantics
Why am I telling you this story?
STOP
STEP BACK
LOOK AROUND

Where is industry?
Industry is Celebrating!

• XML saves the day!
• “Don’t need a header section, a task section, a resources section!”
• “Don’t need to write a parser!”
• “XML eliminates syntax issues and captures all the semantics!”
Reaching Across the Gap

Need to educate

Educate the customer
- Completeness
- Consistency
- Provability
- Type vs. subclass
- Intension vs. extension
- ...

... and be educated

and learn about the customer

- Bubbles and arrows
- Tables
- Spreadsheets
- Strings instead of concepts
Fertile Opportunity

- Existing information exchange standards strive to provide precise definitions of terms and relations.

- ...and usually attempt to do so using English, XML, or UML.
The results are difficult to validate, test for conformance, or test for interoperability
The Smart Electrical Grid

- Currently over 200 standards to be reviewed for the SGIP* Catalog of Standards
  - Overlapping, different, sometimes contradictory vocabularies and definitions

- Need a way to manage
  - Inconsistencies
  - Constraints on usage
  - Relationships between terminologies

*SGIP = Smart Grid Interoperability Panel
Example: “Meter”

IEC 61968

IEC 61970

Multispeak V4.1

NAESB PAP10
IEC 61968

Meter Attributes and Meter Associations

Functionality

Inventory management
Emphasis is on the actions taken on a meter. Meter is not modeled, and is simply identified via a meterID string.

The “meter” class is used to support meter replacement.
Should it be an:
- Object?
- Device?
- Measuring Device?
- Meter?
- Electric Meter?
“date”

Five different root classes
Do any derive from ISO 8601?
Need explicit ancestry
Linking Standards
Declaring Primitive Datatypes

Finding all declared primitive datatypes in the model that inherit from definitions from OMG or W3C (i.e. UML or xsd definitions)
Some of the Areas where Ontological Tools can Help

- Achieving basic consistency in use of terms, data types
- Aligning different levels of abstraction
- Identifying and relating different contexts
- Managing multiple versions of highly interconnected models
- Identifying opportunities for model refactoring
- Enforcing modularity

- Cannot usually just start over with a committee and build a new ontology
Semantic Harmonization of Smart Grid Concepts - SPARQL Query Editor

Sample Queries

1. Datatypes not inheriting from standards
2. Datatypes not inheriting from standards, filtered with a matching string in role
3. Classes defined but never referred to in a relation
4. Classes that share substantially the same properties
5. All Object Properties (roles) containing a string, and trace the ancestry of the range class
6. All Object Properties for a Class containing a string, and trace the ancestry for each Object Property
7. All Object Properties in a namespace containing a string, and trace the ancestry for each Object Property
8. All Properties (roles) containing a string, and trace the ancestry of the range class
9. All datatypes inheriting from standard definitions or the Common Primitive Types
10. All external classes pointing to FSGIM classes
11. Navigate up superclasses to find the ultimate parent class, for all classes containing a given string
12. Number of core component classes
13. Properties redundantly defined by subclasses
14. Searching for properties containing a string
15. Superclasses that have no properties
16. Datatypes not inheriting from standards, filtered with a matching string in declared datatype (range)
17. Relations with cardinality of at least 1 (one) in a namespace containing a string

WHERE {
    ?role a owl:ObjectProperty .
    BIND (afn:localname(?role) AS ?roleLocalName) .
    BIND (fn:lower-case(?roleLocalName) AS ?roleLocalNameLowerCase) .
    FILTER fn:contains(?roleLocalNameLowerCase, "unit") .
    ?restriction owl:allValuesFrom ?range .
    OPTIONAL {
        ?range (rdfs:subClassOf)+ ?rootClassOfRange .
        ?rootClassOfRange a owl:Class .
        ?rootClassOfRange rdfs:subClassOf owl:Thing .
    FILTER (?rootClassOfRange != owl:Thing) .
    }
}

Query Result

<table>
<thead>
<tr>
<th>role</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>61850-cdc--iec61850:hasUnitsRef</td>
<td>61850-cdc--iec61850:UnitsEnumeratedValue</td>
</tr>
<tr>
<td>content_information--iso_19115_metadata:hasUnits0..1Ref</td>
<td>units_of_measure--derived:UomLength</td>
</tr>
<tr>
<td>coordinate_systems--iso_19111_spatial_referencing_by_coordinates:hasAxisUnitIDRef</td>
<td>units_of_measure--derived:UnitOfMeasure</td>
</tr>
<tr>
<td>data_quality_information--iso_19115_metadata:hasValueUnitRef</td>
<td>units_of_measure--derived:UnitOfMeasure</td>
</tr>
<tr>
<td>distribution_information--iso_19115_metadata:hasUnitsOfDistribution0..1Ref</td>
<td>text--primitive:CharacterString</td>
</tr>
<tr>
<td>distribution_information--iso_19115_metadata:hasDensityUnits0..1Ref</td>
<td>text--primitive:CharacterString</td>
</tr>
<tr>
<td>generator_component--model_components:hasDERUnitStatusRef</td>
<td>61850-7-420--iec61850:DERUnitStatus</td>
</tr>
<tr>
<td>generator_component--model_components:hasDERUnitSupportRef</td>
<td>61850-7-420--iec61850:DERUnitSupport</td>
</tr>
</tbody>
</table>
How much is enough?

• Most industrial standards are still emerging from pdf documents and tables
• Some forward-looking standards are in XSD or UML
What are some **minimal** requirements needed to start improving the quality of **existing** standards?

Most standards today

<table>
<thead>
<tr>
<th>Capability</th>
<th>Paper, .doc, .pdf</th>
<th>XML/XSD</th>
<th>UML</th>
<th>OWL</th>
<th>FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine readable</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Queryable</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reasoning</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Proofs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>