A ROBUST PROCESS ONTOLOGY FOR MANUFACTURING SYSTEMS INTEGRATION

Craig Schlenoff, Rob Ivester, Amy Knutilla

National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

In all types of communication, the ability to share information is often hindered because the meaning of information can be drastically affected by the context in which it is viewed and interpreted. This is especially true in manufacturing because of the growing complexity of manufacturing information and the increasing need to exchange this information among various software applications. Different manufacturing functions may use different terms to mean the exact same concept or use the exact same term to mean very different concepts. Often, the loosely defined natural language definitions associated with the terms contain so much ambiguity that they do not make the differences evident and/or do not provide enough information to resolve the differences.

A solution to this problem is the development of a taxonomy, or ontology, of manufacturing concepts and terms along with their respective formal and unambiguous definitions. This paper focuses on an effort at the National Institute of Standards and Technology to identify, formally define, and structure the semantic concepts intrinsic to the capture and exchange of discrete manufacturing process information.

KEYWORDS

Process Specification; Process Ontology; Manufacturing Systems Integration; Ontological Engineering; Formal Semantics; Knowledge Interchange Format (KIF)

1. Introduction and Related Work

As the use of information technology in manufacturing operations has matured, the capability of software applications to inter-operate has become increasingly important. Initially, translation programs were written to enable communication from one specific application to another, although not necessarily both ways. As the number of applications has increased, and the information has become more complex, it has become much more difficult for software developers to provide translators between every pair of applications that need to exchange information. Standards-based translation mechanisms, such as the Standard for the Exchange of Product Model Data (STEP) [1], have simplified translator development for some manufacturing software developers. By developing a single translator between an application that uses product model data and STEP, the application can inter-operate with a wide variety of other applications that have a similar translator between STEP and their application.

In an analogous manner to the role of STEP in exchange of product model information, manufacturing software applications would benefit from a standard for the exchange of process information. Many manufacturing engineering and business software applications use process information, including production scheduling, manufacturing process planning, workflow, business process reengineering, simulation, process realization, process modeling, and project management. Each of these applications utilizes process information in a different way, so it is not surprising that these applications' representations of process information are different as well. The primary difficulty with developing a standard to exchange process information is that these applications sometimes associate different meanings to the terms representing the information that they are exchanging. For example, a resource represented as RESOURCE in two different applications would likely be mapped to one another even if the two applications have a slightly different understanding of the resource. In the case of a workflow system, a resource is primarily thought of as the information that is used to make necessary decisions. In a process planning system, a resource is primarily thought of as a person or machine that will perform a given task. If one were to integrate a process model from a workflow and a process planning application, their first inclination would most likely be to map one resource concept to the other. This mapping would undoubtedly cause confusion. Therefore, both the semantics and the syntax of these applications need to be considered when translating to a neutral standard. In this case, the standard must be able to capture all of the potential meanings behind the information being exchanged.

The Process Specification Language (PSL) project at the National Institute of Standards and Technology (NIST) is creating a neutral, standard language for process specification to serve as an interlingua to integrate multiple process-related applications throughout the manufacturing life cycle. This project is working/has worked closely with other efforts, such as A Language for Process Specification (ALPS) Project [2], the Process Interchange Format (PIF) Project [3], the Toronto Virtual Enterprise (TOVE) Project [4], and the Enterprise Ontology Project [5]. ALPS was a research project at NIST which identified information models to facilitate process specification and to transfer this information to process control. PIF is an interchange format based upon formally defined semantic concepts, like PSL. However, unlike PSL, PIF is focused on modeling business processes and offers a single syntactical presentation, the BNF (Backus-Naur Format) specification of the Ontolingua Frame syntax. The TOVE project provides a generic, reusable data model that provides a shared terminology for the enterprise that each agent can jointly understand and use. The Enterprise Ontology project's goal is to provide "a collection of terms and definitions relevant to business enterprises to enable coping with a fast changing environment through improved business planning, greater flexibility, more effective communication and integration" [5]. While both TOVE and the Enterprise Ontology focus on business processes, there are common semantic concepts in both these projects and the manufacturing process-focused PSL.

The plan for the PSL project has five phases: requirements gathering, existing process representation analysis, language creation, pilot implementation and validation, and submission as a candidate standard. The completion of the first phase resulted in a comprehensive set of requirements for specifying processes [6]. In the second phase, twenty-six process representations were identified as candidates for analysis by the PSL team and analyzed with respect to the phase one requirements [7]. Nearly all of the representations studied focused on the syntax of process specification rather than the meaning of terms, or semantics. While this is sufficient for exchanging information between applications of the same type, such as process planning, different types of applications associate different meanings with similar or identical terms. As a result of this, part of the third phase involves the development of a formal semantic layer (an ontology) for PSL, which is the focus of the remainder of this paper.

2. The PSL Ontology

The foundation of the process specification language is the PSL ontology, which provides rigorous and unambiguous definitions of the concepts necessary for specifying manufacturing processes to enable the exchange of process information. The PSL ontology is represented using the Knowledge Interchange Format (KIF) specification [8]. Briefly stated, KIF is a formal language developed for the interchange of knowledge among disparate computer programs (written by different programmers, at different times, in different languages, and so forth). KIF provides the level of rigor necessary to unambiguously define concepts in the ontology, a necessary characteristic to exchange manufacturing process information using the PSL ontology.[1]

The PSL ontology is essentially two-tiered. The foundation of the ontology (the first tier) is a set of process-related concepts that are common to ALL manufacturing applications that were investigated during the first phase of the project [6]. These concepts constitute the core of the PSL ontology and include concepts such as resources, processes, and time intervals. However, these concepts alone would only allow for the exchange of very simple process specifications. Therefore, this ontology includes a mechanism to allow for extensions to these core concepts (the second tier) to ensure the robustness of the ontology. Initially, these extensions will be derived from a series of pilot implementations designated to demonstrate that the PSL can successfully address real world exchange scenarios.

The PSL ontology is based upon a small set of primitive concepts, *activity*, *object*, *time point*, and *relationship*. Because these are primitive concepts, there are no concepts with which to define them, yet truths or axioms can be stated about them. These truths provide the user enough detail to gain the level of understanding necessary to be able to use the primitive concepts

[1] In addition, KIF is being proposed as a standard to ANSI (American National Standards Institute) by X3T2 (National Committee for Information Technology Standards), Technical Committee T2 (Information Exchange and Interpretation) and is currently in the balloting phase.

effectively. By basing an ontology on primitive concepts, you remove the circularity found in other types of information sources, such as the dictionary, where terms are defined with respect to one another. Example truths provided for the primitive concepts (expressed in KIF) are included below.

Axiom 1. Everything is either an activity, object, or timepoint.

```
(or (Activity ?x) (Object ?x) (Point ?x))
```

Axiom 2. The begin point of every activity or object is before or equal to its end point.

```
(=> (or (Activity ?x) (Object ?x))
(BeforeEq (Beginof ?x) (Endof ?x)))
```

All other definitions of concepts within the PSL ontology can be traced back to these primitive concepts. For example, the concept of *serial tasks* can be traced back to the concepts of *activity*, *time point*, and the relationships between them. This is expressed in the PSL Ontology through the following KIF statements:

Definition 1: A serial activity is any activity that is deterministic and totally ordered.

```
(forall (?a)
        (<=>        (serial ?a)
        (and    (deterministic ?a) (total_ordered ?a))))
```

Definition 2: In a deterministic activity, all of its sub-activities occur at some point whenever the activity itself occurs.

```
(forall (?a ?a1 ?s1 ?s2)
(<=>    (deterministic ?a)
        (=>    (and    (Do ?a ?s1 ?s2) (subactivity ?a1 ?a))
                  (exists (?s3 ?s4)
                          (and    (Do ?a1 ?s3 ?s4) (leq ?s1 ?s3) (leq ?s3 ?s4) (leq ?s4 ?s2))))))
```

Definition 3: In totally ordered activities, the ordering over which sub-activities occur is a total ordering, that is, given any two activities, either one occurs before the other or they occur at the same time.

```
(forall (?a ?a1 ?a2 ?s1 ?s2 ?s3 ?s4)
(<=>    (total_ordered ?a)
                (=>    (and    (subactivity ?a1 ?a) (subactivity ?a2 ?a) (Do ?a1 ?s1 ?s2) (Do ?a2 ?s3 ?s4))
        (or    (leq ?s1 ?s3) (= ?s1 ?s3) (leq ?s3 ?s1)))))
```

With the establishment of an unambiguous ontology for process exchange, it will be necessary to understand what it means to be conformant to PSL, or for that matter, any neutral representation. One could either be conformant to the terminology (i.e., use the same terms that the neutral representation uses) or be conformant to the meaning of the terms (i.e., have the flexibility to use different terms but ensure those terms can map to the meanings expressed in the PSL ontology). For example, a PSL conformant representation could use the words *sequential task* to represent two tasks that occur one directly after the other even though the PSL ontology uses the words *serial activity* to represent the same concept. The PSL defines conformance as agreement in meanings as opposed to agreement in terminology. This allows other representations the freedom of using whichever term is most appropriate for their domain as long as they can map the meaning of that term to a concept within the PSL ontology.

3. Discussion and Conclusion

A community of researchers worldwide is participating in the development of the Process Specification Language. The coordination of this research and development by NIST will ensure its applicability to the open and unambiguous exchange of process information within the manufacturing domain. NIST will also serve as a champion to promote the PSL as an international standard for the exchange of process information.

Other efforts to develop mechanisms for the exchange of data, such as STEP, have focused on syntactical standards elements necessary for data exchange. This focus works well for exchanging information among similar domains where the terms used have the same meanings. However, within the increasingly complex manufacturing environment where process models are maintained in different software applications, standards for the exchange of this information must address not only the syntax but also the meanings or semantics of terms and concepts used. PSL uniquely addresses this in its identification and development of semantics for specifying and exchanging process information. The identification of the necessary concepts was based on a thorough analysis of the requirements for specifying business and manufacturing engineering processes in the manufacturing domain and then analyzing a broad set of existing approaches to representing process models with respect to these requirements.

The next challenge is to develop the methods for mapping between the often-implicit semantics of existing process representations with the explicit and the well-defined semantic concepts of the PSL. This mapping will define the translation of syntactical elements of a representation to and from the PSL. One approach to accomplishing this can be found at http://www.nist.gov/psl/infoexch/. Future plans for the PSL include extending the semantics, syntax, and translation mechanisms within manufacturing scenarios (e.g., exchanging process information among process planning, scheduling, and simulation software applications) and then proposing a validated PSL as a candidate international standard for process information exchange to an appropriate standards body.

REFERENCES

1. ISO, “Product data representation and exchange: Part 1: overview and fundamental principles,” ISO Standard 10303-1, 1992.
2. B. CATRON, S. RAY, “ALPS: A Language for Process Specification”, Int. J. Computer Integrated Manufacturing, Vol. 4, No. 2, 105-113, 1991.
3. J. LEE, et al, “The PIF Process Interchange Format and Framework” to appear in Knowledge Engineering Review 13(1), Cambridge Univ. Press, 1998.

4. M. S. FOX, et al, "An Organization Ontology for Enterprise Modeling: Preliminary Concepts," Computers in Industry Vol. 19, pp. 123-134, 1996.
5. M. USCHOLD et al "The Enterprise Ontology," The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use, 1998.
6. C.I. SCHLENOFF et al., "Unified Process Specification Language: Requirements for Modeling Process," NISTIR 5910, National Institute of Standards and Technology, Gaithersburg MD, September 1996. (also available at http://www.nist.gov/psl/)
7. A. KNUTILLA, et al, "Process Specification Language: Analysis of Existing Representations," NISTIR 6133, National Institute of Standards and Technology, Gaithersburg MD, October 1997.
8. M. GENESERETH, R. FIKES, "Knowledge Interchange Format Version 3.0 Reference Manual," Report Logic-92-1, Stanford CA, June 1992.