Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2000), *14*, 257–270. Printed in the USA.
Copyright © 2000 Cambridge University Press 0890-0604/00 $12.50

An analysis and approach to using existing ontological systems for applications in manufacturing

CRAIG SCHLENOFF, PETER DENNO, ROB IVESTER, DON LIBES, AND SIMON SZYKMAN
National Institute of Standards and Technology, Gaithersburg, MD 20899

(RECEIVED September 1, 1999; ACCEPTED January 31, 2000)

Abstract

This paper reports on the results of an analysis of existing ontological systems to determine which is most appropriate for the manufacturing domain. In particular, this involved the exploration of efforts that are studying both the uses of ontologies in the general sense and those that are using ontologies for domain-specific purposes. Eleven ontological systems were analyzed and, using a set of analysis criteria, it was determined that the Cyc (Cyc is a registered trademark of Cycorp Inc.) system was most appropriate for modeling concepts in the manufacturing domain. After the analysis is described, examples are given to show how manufacturing concepts could be modeled in the Cyc system. This work is part of a larger project whose objective is to move closer to the ultimate goal of seamless manufacturing systems integration using the principle behind ontological engineering to unambiguously define domain-specific concepts. The output of this work will be a taxonomy of manufacturing terms and concepts along with formal definitions of exactly what each of those terms and concepts mean and how they interrelate.

Keywords: Analysis; Inferencing; Manufacturing Ontology; Ontological Systems; Taxonomy

1. INTRODUCTION

As the use of information technology in manufacturing operations has matured, the capability of software applications to interoperate 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 have simplified integration for some manufacturing software developers by requiring only a single translator to be developed between their respective software product and the interchange standard. By developing only this single translator, the application can interoperate with a wide variety of other applications that have a similar translator between that standard and their application.

This challenge of interoperability is especially apparent with respect to manufacturing information. Many manufacturing engineering and business software applications use process information, including manufacturing simulation, production scheduling, manufacturing process planning, workflow, business process reengineering, product realization process modeling, and project management. Each of these applications utilizes manufacturing information in a different way, so it is not surprising that these applications' representations of manufacturing information are different as well. The primary difficulty with developing a standard to exchange manufacturing information is that these applications sometimes associate different meanings with the terms representing the information that they are exchanging. For example, 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 with a process-planning application, one's first inclination would most likely be to map one resource concept to the other. This mapping would undoubtedly cause confusion. Therefore, both the semantics

---

Reprint requests to: Kristy Thompson, National Institute of Standards and Technology, Mail Stop 8260, Gaithersburg, MD 20899-8260, USA. E-mail: kristy@cme.nist.gov

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

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.

To quantify the cost to industry, the cost of interoperability among just process-related applications is a $2 billion market. It is estimated that methodologies and formal specifications to better enable more efficient exchange of manufacturing process information could have a 15–20% cost reduction translating into a $300 million cost savings to industry (Source: Gartner Group Analyst).[1] As another example, an average of six point-to-point translators are written per installation of CAM packages at $6,000 each. Estimating 50,000 installations of CAM software in the United States per year, this translates into $300 million in interoperability cost per year (ignoring interoperability maintenance cost, which totals about 25% of total maintenance cost) (Freimuth, 1998).

The objective of the work described in this paper is to move closer to the ultimate goal of seamless manufacturing systems integration using the principle behind ontological engineering to unambiguously define domain-specific concepts. A major challenge facing industry today is the lack of interoperability between heterogeneous systems. Current integration efforts are usually based solely on how information is represented (the syntax) without a description of what the information means (the semantics). With the growing complexity of information and the increasing need to completely and correctly exchange information among different systems, the need for precise and unambiguous capture of the meaning of concepts within a given system is becoming apparent.

The approach for the project described in this paper is to analyze current ontological systems to determine which is most suitable to model the concepts in the manufacturing domain. Examples of ontological systems include Cyc[2,3,4] Mikrokosmos, and the Ontolingua server. The project is in the process of formally identifying and modeling concepts (and definitions of those concepts) from various manufacturing domains and projects (e.g., process specification, product modeling, etc.) in this ontological system. At this point, an analysis will be performed to help to identify inconsistencies in the use of terms among various domains as well as help to establish a means to generalize these terms to a level that is common among the domains in question.

The output of the work documented in this paper will be a taxonomy of terms and concepts along with formal definitions of exactly what each of those terms and concepts mean and how they interrelate (an ontology). Although it would be impossible to create a complete ontology of every interpretation of every term, a high-level, extensible subset of this ontology will be created to serve as a basis for future, domain-specific additions and specializations. This shared understanding of concepts could then be used to integrate applications and systems that function towards a common goal.

Within the scope of this paper, an *ontology* is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve some set of problems. An ontology necessarily entails or embodies some sort of world view with respect to a given domain. The world view is often conceived as a set of concepts (e.g., entities, attributes, or processes), their definitions, and their inter-relationships; this is referred to as a *conceptualization*.

All ontologies consist of a vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. The various ontologies can also be distinguished by their degree of formality in the specification of meaning. With informal ontologies, the definitions are expressed loosely in natural language. Semiformal ontologies provide weak axiomatizations, such as taxonomies, of the terminology. This can serve as a framework for shared understanding among people, but is often insufficient to support interoperability, and ambiguity can hinder integration. Formal ontologies define the language of the ontology, a set of intended interpretations of the terminology, and a set of axioms which are sound and complete with respect to the intended interpretations, that is, every consistent interpretation of the axioms is an intended interpretation, and every intended interpretation is consistent with the axioms.

This paper documents the results of the first phase of this project—that of analyzing existing ontological systems to determine which is most appropriate for the manufacturing domain. In particular, this phase involved the exploration of efforts that are studying both the uses of ontologies in the general sense and those that are using ontologies for domain-specific purposes.

2. WHY ONTOLOGIES FOR MANUFACTURING?

This section considers what value the investigated ontologies might provide in the area of information technology within manufacturing. Communication and context are important notions in understanding the role of the investigated ontologies with respect to other technologies. This and the related concepts of ground, context availability, seamlessness, and formality are discussed. Two areas of potential benefit are then considered: unambiguous communication, and the future industrial information infrastructure.

---

[1] Personal communication with a Gartner Group analyst, March 1999.

[2] Cyc is a registered trademark of Cycorp Inc.

[3] No approval or endorsement of any commercial product in this paper by the National Institute of Standards and Technology is intended or implied.

[4] This paper was prepared by United States Government employees as part of their official duties and is, therefore, a work of the U.S. Government and not subject to copyright.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

2.1. Communication, meaning and context

In this paper *communication* has the following meaning: One communicates to another with the expectation that at some time thereafter the receiver will produce a behavior that is in some way consistent with the initiator's intention. For communication to succeed, the initiator must have (or the system design must reflect) some understanding of the context under which the receiver is operating and the relationship between the message it designs and the behavior it desires from the receiver. Along the lines of Bloomfield (1933), there is at the very least this sense of meaning to communication: It sets conditions for satisfaction.

Context in the above definition of communication is an environment, a "place" where things occur or an utterance is made. It is largely a matter of how context is established and used that differentiates various software technologies. We can state two fundamental problems of communication as 1) that individuals seldom share a common network of symbols, and 2) that designating a place in a network (establishing a context) is difficult. In essence, all problems of context in software (excepting perhaps some artificial intelligence applications) are supposed to be resolved through an agreement on what the information exchanged means. The software developers might never know or care whether they are developing interfaces based on identical understanding. What is essential is that the software behaves as expected. Much the same can be said of the interactions of people.

2.2. Ground, context availability, seamlessness and formality

The American Heritage Dictionary defines *ground* as "the foundation for an argument, a belief or an action; a basis." Ontologies are one technology where the notion of ground is prominent. In some sense, relating consensus domain terminology to widely held ground terms enables unambiguous communication of information. Some ontology systems, such as Cyc, provide a mechanism to allow reasoning under multiple sets of ground terms (deKleer 1986).

As opposed to ontological systems' explicit use of ground, the exchange of a single property in ISO 10303 (informally known as the STandard for the Exchange of Product model data (STEP)) (International Standards Organization, 1994) technology includes several context-setting references: references to the measurement units used, the various representations of the property, the technical discipline it concerns, and the product lifecycle (Danner, 1997).

Seamlessness refers to the continuity (or "transparency") of the information content with its context-setting information. That is, in a seamless environment the problem solving machinery can get context setting information without appealing to another for service. The principle question here with respect to the investigated ontologies is how to marry the technology to the problem solvers. That is, how should ontologies be interfaced to applications such as schedulers and workflow systems?

At times, formality in ontologies seems to mean the degree to which the ontology resembles mathematical logic. Resemblance to mathematical logic in itself however does not suggest a purpose for formality. We suggest the following: Formality is about making valid inferences (and thus getting expected behaviors), traceability to ground, and enabling computational tools (to manage complexity, etc.).

2.3. Answers to "why ontologies"?

The previous section provides a foundation of ideas concerning communicating systems. Assuming that there are reasonable answers to questions regarding how this technology may be employed, we may still ask what added value the investigated ontologies might provide to the development of a manufacturing information infrastructure. Here we consider the contribution ontologies might make to eliminating ambiguity in communication.

Consider the following issues that arise in pursuit of the goal of unambiguous communication: where the problem of ambiguity resides, and whether or not communicating systems need access to ground terms. The problem with adopting universally accepted meanings is getting individuals in whole industries to agree on the meaning of perhaps thousands of terms such as "part version" and "part revision." This will require a tremendous effort, but it is essential to the goal of unambiguous communication. Standards efforts such as IEC 61360-4, a dictionary of standard terminology for electronic components (International Standards Organization, 1997), are representative of the sort of work that must be done.

Communicating systems rarely need access to ground terms. The exception to this is those few systems that mediate data. The problem of bringing meaning to the exchanged data always leads back to reference to shared understanding. For this reason we conclude that ontologies do not offer significant benefit towards making current information exchange methods more reliable.

Distributed objects, agents, integrated workflow and supply chains are common themes emerging in the development of an industrial information infrastructure. This mode of operation emphasizes *ad hoc* access to objects. This is in contrast to the more traditional approach of organizing systems around the semantics of a shared database. The emerging architecture suggests that *ad hoc* access to shared meta-data and terminology might also prove useful in future information systems. If this turns out to be true, then technology such as the investigated ontologies may prove to be essential in supplying a terminology and meta-data in computational form.

Assessing the value of ontologies (or any other technology) to the problem of communicating manufacturing in-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

formation is, in part, a matter of determining whether it is the simplest possible means to establish a context on exchanged data under which valid inferences (and thus expected behaviors) can be achieved.

3. RELATED WORK

Only a handful of efforts have attempted to model manufacturing-related information in a formal ontology. However, there is a larger amount of work that can be leveraged to make this modeling process easier. Described below are seven such efforts whose output was either an ontology of manufacturing-related concepts or who developed a generic ontology which could lend itself to modeling manufacturing concepts.

3.1. ANSI *Ad Hoc* Committee on Ontology Standards

The goal of the ANSI Ad Hoc Group on Ontology Standards (1998) (associated with the ANSI X3T2 Committee on Ontology Standards) is to merge the upper level ontologies of many of the well-known ontological systems (Cyc, Pangloss, Penman, Wordnet, etc.). An *upper level ontology* is an ontology of the most general conceptual categories. There are a number of such ontologies out in the world that have proved very useful in natural language processing and other AI-oriented applications, as well as in enterprise modeling and database integration. The challenge is that it is difficult to translate between these applications because of the differences in their upper level ontologies. The purpose of the standard will be to provide a sort of ontological baseline to support translation and integration between ontology-based applications, and hopefully also to serve as the starting point for future upper level ontologies.

At the time the analysis was performed, all that was available from this group was a high-level taxonomy of terms without any definition of what the terms meant. It was assumed that the location of any term within the taxonomy was meant to serve as a loose definition of the term. However, because this ontology standard was being adopted by other systems that we were analyzing, such as Cyc, the analysis of those other systems would indirectly allow us to analyze the ontology standard. In addition, because those systems provided additional capabilities that the ontology standard alone did not (e.g., inferencing capabilities, formal definitions of terms, user interfaces, etc.), the respective systems would be a more appropriate choice for use to model a manufacturing ontology. For these reasons, this Ontology Standard was not investigated any further.

3.2. Cyc

Cyc (Cycorp Inc., 1998) is a very large, multicontextual knowledge base and inference engine developed by Cycorp. The goal of the Cyc project is to construct a foundation of basic common sense knowledge—a semantic substrate of terms, rules, and relations—that will enable a variety of knowledge-intensive products and services. Cyc is intended to provide a deep layer of understanding that can be used by other programs to make them more flexible.

A drawback to Cyc is that its level of knowledge is so deep as to be unintuitive to all but Cyc knowledge experts. Higher-level knowledge is left to application developers. Not surprisingly, there are large gaps in Cyc's higher-level knowledge base (KB), as it has only been extended to support whatever application was required for its use. Only some aspects of these extensions are publicly available. Manufacturing is not well represented by the KB.

The Cyc technology is composed of the knowledge base and inference engine, the CycL representation language, interface tools, and application modules. Cycorp is currently working on tools to ease the difficulty of adding to the KB. At the present time, the Cyc KB contains tens of thousands of terms and several dozen hand-entered assertions involving each term. CycL is a large and flexible knowledge representation language. It is essentially an augmentation of first-order predicate calculus (FOPC), with extensions to handle equality, default reasoning, and some second-order features.

3.3. Enterprise Ontology

The Enterprise Ontology (Uschold et al., 1998) was built as part of the large Enterprise Project at the Artificial Intelligence Applications Institute at the University of Edinburgh, in collaboration with industry partners. The focus of the project is to promote the use of knowledge-based systems in enterprise modeling and organizational support. The result of this initiative was an Enterprise Toolset, one component of which is the Enterprise Ontology.

The Enterprise Ontology is relatively comprehensive and includes over 90 different concept classes and over 60 relations between concepts. In order to represent concepts within the Enterprise Ontology itself, a *meta ontology* was developed, which includes more general modeling terms such as entities, relationships, roles, attributes, and so on. Building on these terms, the concepts in the Enterprise ontology are divided into five categories: activities, organization, strategy, marketing, and time. Of course, there are interactions among the various categories of concepts. For example an activity may take place over an interval of time as part of a plan.

The intent of the Enterprise Ontology is not to model specific types of enterprises, but to provide a general model that is oriented more towards business and organization than towards a specific domain. From the perspective of the evaluation being performed in this paper, the Enterprise Ontology is greatly lacking. Virtually all concepts and terms that are specific to manufacturing enterprises are missing from this enterprise model. However, the Enterprise Ontology is still viewed as a valuable resource because of the infrastruc-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

ture it provides. The meta ontology provides a flexible set of primitives for building concepts, and because manufacturing enterprises are a subset of business enterprises in general, many of those aspects of a manufacturing enterprise that are not manufacturing specific are present in the existing ontology. For instance, concepts such as resources, people, machines, and plans will have direct applicability within a manufacturing enterprise model. It should be noted that that in most cases, for application to a manufacturing enterprise, further specification of concepts existing within the current ontology will be necessary.

3.4. LOOM

> “Loom is a language and environment for constructing intelligent applications. The heart of Loom is a knowledge representation system that is used to provide deductive support for the declarative portion of the Loom language. Declarative knowledge in Loom consists of definitions, rules, facts, and default rules. A deductive engine called a classifier utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies in order to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing” (LOOM Homepage, 1998).

As this quote makes clear, Loom is a language and environment. It is not an ontology itself but is quite suitable for implementation of projects using ontologies. Loom is written in Common Lisp and the Common Lisp Object System (CLOS) and is easily integrated into Common Lisp programs. The importance of Loom in this study is that it exemplifies the sort of infrastructure that exists to enable development of high-quality knowledge-based systems. Because Loom is not a commercial product (it is the intellectual property of the University of Southern California) there are fewer barriers to its use.

Our exploratory work with Loom suggests that it is easy to use. Although there may be concerns among some about Common Lisp not being a mainstream programming language, the development of a robust Common Lisp-based HTTP server and a CORBA binding to Common Lisp has eased this problem somewhat.

3.5. Mikrokosmos

The ultimate objective of the Mikrokosmos (Mahesh & Nirenburg, 1995) research project is to define a methodology for representing the meaning of text in a language-neutral format called a text meaning representation (TMR). This would provide a mechanism for Knowledge-Based Machine Translation (KBMT) of natural language text from one language to another (via an intermediary translation into a TMR). In pursuit of this goal, researchers at New Mexico State University have conducted a comprehensive study of linguistic and language use phenomena. These phenomena have been encapsulated in various microtheories that are united through the control architecture of the KBMT system.

The principle objective of the Mikrokosmos project is, unfortunately, not directed at arbitrary queries of a specific knowledge base, but rather, a general mechanism for mapping meaning between languages. As such, it has been developed with different capabilities and design structure than would be needed for a manufacturing ontology. Specifically, Mikrokosmos provides no inferencing capability for answering questions that are not explicitly answered in the knowledge base. This capability is vital for providing useful information in a manufacturing context. The Mikrokosmos ontology contains a wide variety of basic concepts related to manufacturing (e.g., drill, cut, and make), but it has very few detailed concepts that would be helpful for manufacturing. As such, implementing a manufacturing ontology using Mikrokosmos would require the development of tools for inferencing capabilities and general querying of the knowledge base, as well as adding a tremendous number of detailed concepts to the knowledge base.

3.6. Ontolingua

The Ontolingua (Farquhar et al., 1997) ontology development environment, developed at the Stanford University Knowledge Systems Laboratory, consists of a suite of authoring tools for creating and browsing modular, reusable ontologies. The set of tools provides a World Wide Web-based interface for ontology creation, allowing remote ontology creation or browsing of existing ontologies, many of which are available through the server Ontolingua Server at Stanford University.

The Ontolingua ontology development environment models information using the Ontolingua language (Gruber, 1995), a language based on the knowledge interchange format (KIF) (Genesereth & Fikes, 1992). Ontolingua expands the basic first-order predicate logic formalism provided by KIF to also include syntax for an object-oriented representation (classes, instances, slots, relations, etc.) In addition to the web-based authoring interfaces, the development environment also provides translation into other knowledge representation languages, including Loom (MacGregor, 1991), Epikit (Genesereth, 1990), Generic-Frame (Chaudhri et al., 1997) and pure KIF.

The purpose of this paper is to evaluate ontologies and not ontology authoring tools. Because this body of work is a development environment, it is not appropriate to attempt to evaluate its direct applicability to manufacturing. However, because of its advantages (ease of use, availability of existing modular ontologies to leverage from, ties to KIF, and translator facilities to interface with other knowledge representation languages), this environment would be a strong candidate for consideration if a new manufacturing-related ontology were to be built from scratch. Indeed, this development environment was used to model the Enter-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

prise Ontology, which is one of the ontologies evaluated in this paper.

3.7. TOVE (TOronto Virtual Enterprise)

To support enterprise integration, it is necessary that a shareable representation of knowledge be available that minimizes ambiguity and maximizes understanding and precision in communication. Secondly, the creation of such a representation should eliminate much of the programming required to answer simple common-sense questions about the enterprise. The goal of the TOVE project (Gruninger, 1998) is to create a generic, reusable data model that has the following characteristics:

- provides a shared terminology for the enterprise that each agent can jointly understand and use,
- defines the meaning of each term (*semantics*) in a precise and as unambiguous a manner as possible,
- implements the semantics in a set of axioms that will enable TOVE to automatically deduce the answer to many common-sense questions about the enterprise, and
- defines a symbology for depicting a term or the concept constructed thereof in a graphical context.

The TOVE reusable representation represents a significant ontological engineering of industrial concepts. All axioms and definition are specified natively in the KIF (Genesereth & Fikes, 1992). It also has presentations using the frame ontology from the Knowledge Systems Laboratory (KSL) (http://www.ksl.stanford.edu/) from Stanford and will shortly have a presentation in XML (eXtensible Markup Language) (Extensive Markup Language, 1998).

The work began by translating the ontologies developed at Carnegie Mellon from LISP into a C++ environment. The ontology was then modified and extended. Currently, the ontology spans: activities, state, causality, time, resources, inventory, order requirements, and parts. There has also been work to axiomatize the definitions for portions of our knowledge of activity, state, time, and resources. The axioms are implemented in Prolog and provide for common-sense question answering via deductive query processing. Future work will focus on the development of ontologies and axioms for quality, activity-based costing, and organization structures.

4. APPROACH AND MAJOR FINDINGS FOR MANUFACTURING ANALYSIS

A systematic approach was taken throughout this project to ensure that a proper cross-section of manufacturing-related ontological systems were chosen, appropriate analysis criteria were determined, and a proper analysis was performed. The project started by doing a literature survey to determine what appropriate ontological systems were available. This survey included a thorough search of the web and numerous interactions with colleagues in the ontology field. From this survey, the following ontological systems were identified:

- ANSI Ad Hoc Group on Ontology Standards Representation (ANSI *Ad Hoc* Group on Ontology Standards, 1998)
- Cyc (Cycorp Inc., 1998)
- Enterprise Ontology (Uschold et al., 1998)
- LOOM (LOOM Homepage, 1998)
- Mikrokosmos (Mahesh & Nirenburg, 1995)
- Ontolingua (Farquhar et al., 1996)
- Sensus (Natural Language Group, 1998)
- SPAR (Shared Planning and Activity Representation) (Tate, 1998)
- STEP (Standard for the Exchange of Product model data) (International Standards Organization, 1994)
- TOVE (Toronto Virtual Enterprise) (Gruninger, 1998)
- Wordnet (Cognitive Science Laboratory, 1998)

A high-level analysis of each of the above ontological systems was performed and a few systems were eliminated due to their lack of appropriateness to this project. In general, the project analyzed these ontologies against the following three criteria:

- the ontology's ability to represent manufacturing information (e.g., time-varying concepts, flow of materials, constraints, etc.),
- the amount of manufacturing information that was already represented in the ontology,
- the ability of the ontology to inference over the information represented.

The following systems were excluded from the analysis, along with the respective reason:

- ANSI Ad Hoc Committee on Ontology Standards—at the time the analysis was performed, this ontology was not mature enough to properly analyze. In addition, because the upper level of Cyc was to be merged with this ontology, an analysis of Cyc would be sufficient to analyze this ontology also.
- Sensus—only a taxonomy of terms without definitions was provided and the concepts represented in this system had already been merged with Cyc through the Ad Hoc Group on Ontology Standards work.
- SPAR—at the time this analysis was performed, it was not mature enough to analyze.
- STEP—it was too limited in domain (only product data), there were no formal definitions of concepts, and from the project participants' previous work with STEP, we know it would not be appropriate.
- Wordnet—it is more of an on-line super-dictionary than a knowledge base.

Table 1 summarized the major points related to the ontologies that were investigated. Once the ontological systems to be analyzed were determined, we moved on to determining the appropriate analysis criteria. It was de-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

Table 1. *Summary of ontologies investigated*

Ontology	Domain	Purpose	Provides Inferencing?	Development Framework or Full Ontology
Cyc	Generic	Enable common sense reasoning about the world	Yes	Full Ontology
Enterprise Ontology	Business enterprise and organization modeling	Comprehensive ontology whose main groupings consist of activities, organization, strategy, marketing, and time.	No	Full Ontology
LOOM	Generic	A language and environment for constructing intelligent applications	Yes (forward, truth maintenance)	Development Framework
Mikrokosmos	Knowledge-based translation of natural language	Translate natural language text from one language to another via a language-neutral text meaning representation	No	Full Ontology
Ontolingua	Generic	Development environment and authoring tool for the creation of modular, reusable ontologies.	No	Development Framework
TOVE	Enterprise integration	Provide a generic, reusable data model including shared terminology and meaning that each agent can jointly understand and use	Yes	Full Ontology

cided that the project would base our analysis on typical manufacturing scenarios. This would involve identifying appropriate manufacturing scenarios, extracting the concepts inherent to that scenario, grouping the concepts into appropriate categories, and developing inferencing questions that are based on those concepts. We would then see how well existing ontological systems could model those concepts and determine how well they could answer questions pertaining to those concepts.

The CAMILE "Factory from Hell" scenario (McKay, 1991) was identified as an appropriate scenario for our manufacturing analysis. This scenario was developed by Ken McKay as part of an assignment through CAM-I (Consortium for Advanced Manufacturing, International) . The scenario details a fictitious factory (based heavily on knowledge gained through site visits to actual factories) including information on many departments and the decision-making processes which occur throughout the development of a product. The concepts, which were detailed in the scenario, were extracted and grouped into manufacturing-related categories. The chosen categories were (in no particular order):

a. Penalties,
b. Costs,
c. Financials,
d. Scheduling,
e. Process planning,
f. Product configuration,
g. Resource planning,
h. Resources,
i. Inventory,
j. Batches/lots,
k. Orders,
l. Customer/vendor,
m. Scrap/rework,
n. Manufacturing execution.

Using these categories and the concepts in each category, we initially examined each of the ontological systems to determine how well they could represent those concepts (see Appendix). We rated each ontology with respect to the following four categories:

1. Required concepts are not represented in ontology. Related information infrastructure is not available and must be modeled before concepts can be represented.
2. Required concepts are not represented in ontology. Related infrastructural concepts are available. Modeling of required concepts could take place primarily by combination of existing concepts.
3. Representation of required concepts could be achieved through specialization or minor modification of existing concepts.
4. Required concepts are available in ontology and would require either trivial modifications or none at all.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

During the initial phases of this analysis, it was found that a few other ontologies were not appropriate for further analysis for the reasons described below.

- LOOM—it is a language and environment. It is not an ontology itself but is quite suitable for implementation of projects using ontologies. Therefore, LOOM would not be appropriate for the development and modeling of a manufacturing ontology.
- Mikrokosmos—its purpose is to provide a general mechanism for mapping meaning between languages. As such, it has been developed with different capabilities and design structure than would be needed for a manufacturing ontology. Specifically, Mikrokosmos provides no inferencing capability to answer questions that are not explicitly answered in the knowledge base.
- Ontolingua—it is an ontology authoring tool and not an ontology itself. Because this body of work is a development environment, it is not appropriate to attempt to evaluate its direct applicability to manufacturing.

For the above reasons, these ontologies were not further analyzed.

The remaining three ontologies, Cyc, Enterprise Ontology, and TOVE, were then analyzed in further detail. The results of this analysis showed that all three packages were approximately equally able to represent manufacturing information. However, the inferencing capabilities in Cyc seemed a bit more mature than the other two packages analyzed. Also, the close relationship that the National Institute of Standards and Technology (NIST) and the Advanced Technology Program Ontology project have with Cycorp would allow the project to more easily leverage Cycorp staff's expertise while modeling the manufacturing ontology. For these reasons, the project decided to proceed with Cyc to model the manufacturing ontology.

5. APPROACHES ON MODELING MANUFACTURING CONCEPTS IN CYC

As distributed by Cycorp, Cyc understands only a few manufacturing-specific concepts. In comparison, the size of STEP, a family of manufacturing-related schemas and related information, suggests that Cyc could be augmented profitably. For our project, using Cyc thus meant starting from a clean slate. There are, of course, many existing bodies of manufacturing standards and draft standards and de facto standards. For our purposes, we chose to draw on STEP and the NIST Process Specification Language (PSL) work (Schlenoff et al., 1996). Specifically, ISO 10303 Application Protocol 213 (Numerical Control Process Plans for Machined Parts) (International Standards Organization, 1995*a*) and Part 49 (Process Structure and Properties) (International Standards Organization, 1995*b*). PSL is a proposed language for process specification.

It is apparent that the manufacturing concepts in Cyc are placeholders to indicate areas of future work rather than the beginning of a manufacturing ontology. On the other hand, Cyc has common-sense knowledge for generic infrastructure that is necessary for manufacturing but not specific to it. As examples, Cyc understands such concepts as sets, overlap, and activities. These should obviously be used in modeling, for example, a model of manufacturing scheduling.

Indeed some concepts found in Cyc map directly to those in PSL. For example, consider *state* (PSL) and *staticSituation* (Cyc). But for the names, these represent the same idea. This is carried forth in related concepts such as *post_state* (PSL) and *postSituation* (Cyc). To understand the differences between different mappings, we went through several exercises comparing different ontological systems that were nominally intended to model the same information. We found minimal overlap in the base knowledge and large gaps of knowledge. Few terms lent themselves to being declared one-to-one.

As an example, here is a PSL constraint defining the idea of irreversibility in terms of state change.

```
(defrelation irreversible (?f) :=
(forall (?s1)
  (⇒ (holds ?f ?s1)
     (not (exists (?s2 ?a)
       (and (not (holds f (do ?a ?s2)))
              (< s (do ?a1 ?s2))))))))
```

In PSL, this would be interpreted as "a state is irreversible if and only if whenever the state holds, then there does not exist a future state where it does not hold."

Cyc defines *StaticSituations* in similar terms. For example, the following Cyc formula defines whether a mental state is also static.

```
(implies
  (and
    (isa ?MENTALSITUATIONFN-1 Mental-
     Situation)
    (isa ?PROP-SLOT PropositionalAt-
     titudeSlot)
    (termOfUnit ?MENTALSITUATIONFN
    (MentalSituationFn ?PROP-SLOT
     ?AGENT ?PROP))
    (termOfUnit ?MENTALSITUATIONFN-1
    (MentalSituationFn ?PER-SLOT
     ?AGENT ?PROP)))
  (isa ?MENTALSITUATIONFN Static-
   Situation))
```

However, many terms in PSL and STEP have no direct analogs. For example, *set_contention* is a PSL function that defines whether there are conflicts if set members are used in multiple activities at the same time. Cyc has no such concept and it must be built from the Cyc primitives that deal with sets (e.g., *SetOrCollection*, *disjointWith*) and activities (e.g., *EventParticipantStatus*).

By no means should this discussion leave the reader with the idea that mapping PSL (or for that matter, any ontolog-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

ical system) to Cyc is straightforward. Correct modeling depends in large part on what deductions you want Cyc to make. In addition, Cyc's massive knowledge of commonsense reasoning suggests that many obvious facts should not be stated explicitly but may be deduced implicitly. In some cases this is counter to the modeling that has been produced in the STEP community, which has no common-sense infrastructure to draw upon.

Microtheories are groupings in Cyc of information and inferencing that is specific to a particular domain. As an example, an engineer may wish to draft a crude schedule for creating rough initial estimates. For simplicity, it would be appropriate to ignore basic knowledge in certain areas. Later, complete but more expensive reasoning can be used for refining schedules. It is clear to us that some types of manufacturing-specific knowledge may belong in different Cyc microtheories.

Creating these microtheories is a second task that we must address in our project—deciding which pieces of knowledge are or are not suitable for use in other areas. This has turned out to be very hard. As an example, the concept of the cost related to the process of scheduling itself is not a generic concept. However, neither is scheduling cost restricted solely to manufacturing, as it arises in anything in which scheduling is hard, such as updating a screen, which is purely in the realm of computer science.

6. DISCUSSION

The analysis presented in previous sections provides an evaluation for several approaches to ontology development, as well as a generic approach toward performing evaluations of ontological frameworks in the context of manufacturing. Aside from evaluating content, as was the primary focus above, the issues of context and inferencing also impact the ultimate utility of ontologies for specific applications.

The main objective of ontology development is to develop a standard vocabulary or to predefine terminology to facilitate exchange of information. Ontologies help create a uniform basis for information exchange by enabling the representation and communication of the meaning of a given term. However, a secondary issue that must be addressed arises when a term has multiple definitions. Being able to represent these definitions formally does not solve the problem of knowing which definition to use in a given circumstance.

This problem is being addressed in several ontology development efforts through the use of contexts in ontologies (see, e.g., Mahesh & Nirenburg (1995); Cycorp Inc (1998); Gruninger (1998)). Context, also referred to in some efforts as microtheories, allows additional information beyond specific formal term definitions to be incorporated into an ontology. This contextual information may be represented implicitly or explicitly within an ontology.

In the case of the Ontolingua Ontology Development Environment, modular ontologies are created and combined or included as components of larger ontologies. In one sense, this can be thought of as an implicit representation of context, since a term may be defined one way in one ontology and differently in another. Mikrokosmos uses context to help resolve the meaning of words that could have multiple meanings. Although the way in which this is accomplished is not entirely disclosed (possibly because it provides them with part of their proprietary advantage), the method involves the use of grammatical rules (e.g., adjectives follow nouns in Spanish, adjectives precede nouns in English). The placement of these words in a sentence provides the context to help to define what the words mean. Cyc represents context using microtheories, each of which is essentially a bundle of assertions that share a common set of assumptions. Typically microtheories are focused on a particular domain of knowledge, a particular level of detail, a particular interval in time, and so forth.

Inferencing, in general terms, is the ability of a system to deduce new information that is not explicitly represented in a knowledge base from concepts that are represented. For example, assume that a particular manufacturing process (Process B) must be performed within 24 h of the completion of another manufacturing process (Process A). For a scheduling program to decide when to schedule Process A, it must have access to certain information. Some of this information would be explicitly represented, such as the expected durations of Processes A and B, the current time, and the standard hours that the factory is open. However, some of the information necessary is unlikely to be explicitly represented, such as whether or not the factory is open tomorrow. This type of information would need to be deduced from information that is explicitly represented, such as, today's date, today's day of the week, scheduling holidays, and factory hours. An inference engine could provide this deductive capability to determine information that is needed but not explicitly represented.

In the ontologies investigated, the tools were designed to work with specific representations; namely: 1) inference engines developed by Cycorp Inc. to work on their CycL representation, 2) a deductive engine developed with LOOM to specifically work on the LOOM knowledge representation (discussed briefly in Section 3.4), and 3) a set of tools developed all around the world to operate on information represented in the KIF.

7. SUMMARY AND AREAS FOR FUTURE RESEARCH

The growing reliance on software tools for capture and exchange of engineering knowledge is bringing about significant benefits in engineering industry. However, with these benefits comes an increasingly significant barrier to further gains in productivity: the difficulty in exchanging formal knowledge with others. As the number of computer-aided manufacturing (CAM) tools and their coverage of the spec-

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

trum of manufacturing activities increases, so too does the impact of this issue.

This paper presents an analysis of existing ontological frameworks from the perspective of exchange of knowledge and information in manufacturing domains. The analysis, summarized in Table 1, covered ontology development frameworks as well as specific instances of ontologies, both domain-specific as well as generic domain-independent ones.

The work presented in this paper comprises a background study performed as part of a project at NIST in the area of ontologies for manufacturing and a related project with the goal of developing a generic PSL. A comparison of concepts contained within Cyc and PSL identified a number of concepts that had virtually identical definitions, but also many concepts that were not shared between the two. In a few cases, definitions in one did not quite match definitions in the other, but more often, a concept in one was not even present in the other. It becomes clear that evaluating the suitability of a given ontological approach for a particular domain such as manufacturing is only a first step. To use an ontology as a knowledge exchange mechanism from one formal representation to another, it is necessary to have a significant overlap in coverage of concepts on both sides. It is at this point that further concept modeling becomes necessary to overcome barriers to effective use.

Ongoing work is addressing the next level of detail with respect to manufacturing knowledge. As part of this work, a preliminary PSL implementation has been used as an interlingua to exchange knowledge between two manufacturing applications (specifically, from PROCAP, a manufacturing process modeling application, to ILOG, a scheduling application). This experiment successfully demonstrated a knowledge exchange between two different representations, but again underscored the need for overlap, not necessarily in terminology, but in concept coverage between applications.

Future work will further investigate the use of ontologies in manufacturing applications by examining the approaches taken toward representation of process, initially in STEP and PSL, and eventually in representations used during a broader set of manufacturing activities. Based on the outcome of the analysis documented in this paper, Cyc has been selected as the modeling framework for subsequent research efforts. As part of these efforts, Cyc will be used for modeling and reconciling concepts that are common to both STEP and PSL, and also to extend beyond manufacturing *process* information to a more general manufacturing ontology.

REFERENCES

ANSI *Ad Hoc* Group on Ontology Standards. (August 20, 1998). http://www.ksl.stanford.edu/.

Bloomfield, L. (1933). *Language*. University of Chicago Press, New York.

Chaudhuri, V., Farquhar, A., Fikes, R., Karp, P., & Rice, J. (1997). The generic frame protocol 2.0, Technical Report KSL-97-05, Stanford Knowledge Systems Laboratory, Stanford, CA.

Cognitive Science Laboratory. (November 30, 1998). Wordnet—A lexical database for English. Princeton University, http://www.cogsci.princeton.edu/~wn/.

Cycorp Inc. (July 31, 1998). The CYC Technology. http://www.cyc.com/tech.html.

Danner, W.F. (1997). Developing Application Protocols using the architecture and methods of STEP Fundamentals of the STEP Methodology. NIST Internal Report (NISTIR) 5972, National Institute of Standards and Technology, Gaithersburg, MD.

deKleer, J. (1986). An assumption-based truth maintenance system. *Artificial Intelligence Journal 28*, 163–196.

Extensible Markup Language (XML) 1.0: W3C Recommendation from 10-February-1998. (August 28, 1998). http://www.w3.org/TR/REC-xml.

Farquhar, A., Fikes R., & Rice. J. (1997). The Ontolingua Server: A tool for collaborative ontology construction. *International Journal of Human Computer Studies*, *46(6)*, 707–728.

Freimuth, K. (1998). Process Speicification Language, A Justification, presentation given at the Process Information Technology: From Research to Industry workshop, March 12–13, 1998, Gaithersburg, MD.

Genesereth, M.R. (1990). *The Epikit Manual*. Epistemics, Inc., Palo Alto, CA.

Genesereth, M.R., & Fikes, R. (1992). Knowledge interchange format, version 3.0 reference manual. Technical Report KSL-92-86, Stanford Knowledge Systems Laboratory, Stanford, CA.

Gruber, T.R. (1995). Toward Principles for the Design of Ontologies Used for Knowledged Sharing, *International Journal of Human and Computer Studies, 43(5/6)*, 907–928.

Gruninger, M. (August 19, 1998). Enterprise Modeling. http://www.ie.utoronto.ca/EIL/comsen.html.

International Standards Organization. (1994). Product data representation and exchange: Part 1: Overview and fundamental principles. ISO, 10303-1:1994.

International Standards Organization. (1995). Product data representation and exchange: Part 213: Application Protocol: Numerical control process plans for machined parts. ISO 10303-213:1995.

International Standards Organization. (1995). Product data representation and exchange: Part 49: Integrated generic resources: Process structure and properties. ISO 10303-49:1995.

International Standards Organization. (1997). Standard data element types with associated classification scheme for electric components. http://www.codus.co.uk/intstds.htm. ISO/IEC 61360-4:1997.

LOOM Homepage. (August 20, 1998). http://www.isi.edu/isd/LOOM/LOOM-HOME.html.

MacGregor, R. (1991). The evolving technology of classification-based knowledge representation systems. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge* (Sowa, J., Ed.), pp. 385–400. Morgan Kaufmann, San Mateo, CA.

Mahesh, K., & Nirenburg, S. (1995). A situated ontology for practical NLP. *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95)*. (also available at http://crl.nmsu.edu/SN.bibliography/SN.papers/ijcai95.pdf)

McKay, K. (1991). Report: Intelligent manufacturing management program, state of the art scheduling survey. CAM-I Report R-91-IMM-01. Consortium for Advanced Manufacturing International (CAM-I), Arlington, TX.

Natural Language Group at USC/ISI. (November 30, 1998). Ontology Creation and Use—SENSUS. http://www.isi.edu/natural-langauge/resources/sensus.html.

Schlenoff, C., Knutilla, A., & Ray, S. (1996). Unified process specification language: Requirements for modeling process NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD. (also available at http://www.nist.gov/psl/)

Tate, A. (November 30, 1998). Planning initiative: Shared planning and activity representation: SPAR homepage: http://www.aiai.ed.ac.uk/~arpi/spar/

The American Heritage Dictionary of the English Language, Third Edition, 1996, Houghton Mifflin Company.

Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1998). The Enterprise Ontology. *The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use*, 31–89.

Varela, F.J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. The MIT Press, Cambridge, MA.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

APPENDIX: SUMMARY OF ANALYSIS OF MOST PROMISING ONTOLOGICAL SYSTEMS

Included in this appendix is a summary of the analyses performed on the most promising ontologies relating to modeling manufacturing information. Each category of manufacturing concepts identified in the analysis criteria (as listed in Section 3.) was given a rating from 1 to 4 (as listed in Section 3.) with respect to each ontology analyzed. A brief explanation for the rating is also provided.

A1. Cyc analysis

General Comments: In Cyc, the philosophy is not to be able to represent a concept with a single predicate or object but rather to be able to do the kinds of reasoning. For instance, there is no object for "addition" but Cyc clearly can reason about addition (as well as do addition). In that sense, the idea that representation could be achieved through "minor modification" or "trivial modification" is not necessarily desirable even though that is clearly implied by the proposed categorizations. This is much like the idea that you cannot dissect the human brain in hopes of finding the "Chicago" neuron (or the neuron for whatever concept you're looking for).

a. Penalties
   Rating: 4
   Comments: Cyc understands penalties and the objects to which they might refer such as vendors, customers, products, sales, prices, etc.

b. Cost
   Rating: 3
   Comments: Cyc understands Cost. Only trivial additions are necessary for it to apply to concepts such as resources and materials, prices, vendors, customers, and sales (which already exist).

c. Financials
   Rating: 2
   Comments: Cyc has no specific concept for financials but appears to be able to do financial reasoning and modeling anyway. Minor additions could prove helpful.

d. Scheduling
   Rating: 4
   Comments: Cyc has an excellent model for scheduling, events, plans, and related concepts.

e. Process planning
   Rating: 3
   Comments: Recent work in modeling process-planning information and functionality could prove useful.

f. Product configuration
   Rating: 3
   Comments: Cyc understands products, product features, and markets. Domain-specific information or classification varies highly.

g. Resource planning
   Rating: 3
   Comments: Resource allocation and planning seems to have an acceptable infrastructure in Cyc. Certain key ideas seem to have no analog but likely (hopefully?) that is the authors' own misunderstanding, as the ontology is somewhat opaque here.

h–i. Resources/Inventory
   Rating: 3
   Comments: Cyc has an excellent provision for high-level resource management. Cyc does not seem to understand its application to manufacturing, so some work is needed on our part.

j. Batches/Lots
   Rating: 3
   Comments: Cyc models these concepts, albeit using completely different words than the industrial engineer.

k. Orders
   Rating: 4
   Comments: Cyc has a very thorough model for order-related concepts (including customers, prices, vendors, organizations, etc).

l. Customers/Vendors
   Rating: 4
   Comments: Same as above (Orders).

m. Scrap/Rework
   Rating: 2
   Comments: Cyc understand the concepts of "throwing something away" and how it might affect the state of the world but rework, recycle, etc all seem to be overlooked by Cyc. Still this doesn't look like it should require a lot of "rework."

n. Manufacturing execution
   Rating: 2
   Comments: Manufacturing-specific information does not exist. However, the generic infrastructure for things like processes, execution, influence factors, resources, resource allocation, capability, all exist.

A2. Enterprise Ontology analysis

Notes: very few axioms exist, as most axioms would not be general but would be domain specific. Thus, even in the best case, additional information modeling will be required for a specific application such as a model manufacturing factory.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

This will probably be true for most ontologies, unless they have already been developed with a specific application in mind.

a. Penalties
 Rating: 2
 Comments: Concepts such as vendors, customers, products, sales, prices, etc. already exist within the ontology. Other concepts that would be required as part of the infrastructure to model penalties also exist, such as time (deadlines might influence penalties), activities and execution of activities, events, etc.

b. Cost
 Rating: In some cases 3, in some cases 4.
 Comments: For costs of the product itself, and possibly costs of resources and materials, prices, vendors, customers, and sales, already exist. For more abstract kinds of costs, additional concept modeling may be necessary if those concepts do not exist.

c. Financials
 Rating: Same as for (b), but in some cases possibly even 2, because financials is much broader than cost.

d. Scheduling
 Rating: 4.
 Comments: Time, events, execution of events, resources, resource allocation and substitution plans, sub-plans, process specification all exist, as do influence factors and assumptions.

e. Process planning
 Rating: Varies from 2 to 4.
 Comments: Maybe we should be more precise about what we mean by process planning.

f. Product configuration
 Rating: 1,2.
 Comments: Products, features, markets, (market) needs exist. The next level of detail that would allow representation of product structure, more general PDM-type concepts, part classifications does not exist.

g. Resource planning
 Rating: 3,4.
 Comments: Planning, resources, resource allocation and substitution, process specification, capability, etc. all exist. Activities exist, but specific activities such as maintenance and repairs do not.

h–i. Resources/Inventory
 Rating: 3.
 Comments: Basic concepts exist, but specialization is required for this application. Things like fixtures, tooling, material, repairs, etc. do not exist.

j. Batches/Lots
 Rating: 3 with a little 2 thrown in.
 Comments: Same as (h–i), but some more basic concepts may not be available, such as splitting (for splitting of batches).

k. Orders
 Rating: 3 with a little 2 thrown in.
 Comments: Customers, prices, vendors, organizations exist. Because this is broader, though, some things may not exist.

l. Customers/Vendors
 Rating: 4.

m. Scrap/Rework
 Rating: 2.
 Comments: Processes, execution, influence factors, etc. exist, but no concepts exist at the level of detail of scrap, rework, associated causes, part evaluation or testing, etc.

n. Manufacturing execution
 Rating: 3.
 Comments: Processes, execution, influence factors, resources, resource allocation, capability, etc. all exist. More specific related concepts do not.

A3. TOVE analysis

a. Penalties
 Rating: 2
 Comments: Although concepts such as delays and resource consumption are defined, there are no explicit concepts for penalties.

b. Cost
 Rating: 4
 Comments: For details on the TOVE Cost Ontology, see "A Cost Ontology for Enterprise Modeling" at http://www.ie.utoronto.ca/EIL/papers/abstracts/30.html

c. Financial
 Rating: 2
 Comments: Work within TOVE has so far concentrated on activity-based costing rather than financial concepts.

d. Scheduling
 Rating: 4
 Comments: For details on the TOVE Scheduling Ontology, see "Intelligent Scheduling Research Group" at http://www.ie.utoronto.ca/EIL/Scheduling.html and "Scheduling Ontology" at http://www.ie.utoronto.ca/EIL/tove/scheduling.html. The following concepts are currently not explicitly defined (and hence have a rating of 3):

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

- Queues,
- Priority.

e. Process planning
   Rating: 4
   Comments: For details on process planning concepts within TOVE, see "Material Flow Ontology" at http://www.ie.utoronto.ca/EIL/tove/material.html.

f. Product configuration
   Rating: 4
   Comments: For details on the TOVE Product Ontology, see EIL Publications on Design the papers at http://www.ie.utoronto.ca/EIL/DITL/design-papers.html. The following concepts are not currently covered by the TOVE Product Ontology, and hence would have a rating of 2 or 3:

- Effectivity,
- "as designed", "as built", "as maintained."

g. Resource planning
   Rating: 4
   Comments: For details, see the "Resource Ontology" at http://www.ie.utoronto.ca/EIL/tove/resource.html and "Material Flow Ontology" at http://www.ie.utoronto.ca/EIL/tove/material.html. The following concepts are currently not explicitly defined in the TOVE ontologies (and hence have a rating of 3):

- Resource preventative maintenance and repairs,
- Resource fixtures and tooling.

h. Resources
   Rating: 4
   Comments: For details, see the "Resource Ontology" at http://www.ie.utoronto.ca/EIL/tove/resource.html. The following concepts are currently not explicitly defined in the TOVE ontologies (and hence have a rating of 3):

- Resource fixtures and tooling.

i. Inventory
   Rating: 4
   Comments: For details, see the "Inventory Ontology" at http://www.ie.utoronto.ca/EIL/tove/inventory.html and the "Material Flow Ontology" at http://www.ie.utoronto.ca/EIL/tove/material.html.

j. Batches/Lots
   Rating: 3
   Comments: These concepts would be an extension of the Resource and Material Flow Ontologies.

k. Orders
   Rating: 4
   For details, see the "Goals Ontology" at http://www.ie.utoronto.ca/EIL/tove/goals.html.

l. Customers/Vendors
   Rating: 2/3
   Comments: The concepts of customer and supplier are defined, but the other concepts in the list (such as synchronization, communication, and meta-issues) are not defined.

m. Scrap/Rework
   Rating: 3
   Comments: Initial concepts can be found in Henry Kim's work on the Quality Ontology as well as the "Material Flow Ontology" at http://www.ie.utoronto.ca/EIL/tove/material.html.

n. Manufacturing execution
   Rating: 2/4
   Comments: Concepts such as tracking, preemption, and iteration are defined (and have a rating of 4), but concepts such as priority, change, and human intervention are not defined (and have a rating of 2).

Craig Schlenoff is a mechanical engineer and the program manager of the Process Engineering Program in the Manufacturing Systems Integration Division (MSID) at the National Institute of Standards and Technology (NIST). The Process Engineering Program's goal is to develop standard representations of process to allow companies to precisely document their manufacturing processes and share these processes among different functions both within their organization and among partnering organizations. Mr. Schlenoff's expertise and interests includes process representation, ontological engineering, information management, and systems integration.

Peter Denno is a computer scientist at NIST. His research interests are information modeling, integration architectures, and product data management.

Robert W. Ivester started at NIST as a National Research Council Postdoctoral Research Associate in September 1996, and is currently working as a Mechanical Engineer in the Automated Production Technology Division (APTD). In between, Dr. Ivester served as Scientific Advisor to the Director of the Manufacturing Engineering Laboratory (MEL) at NIST, and as Special Advisor to the Chief of the APTD. Dr. Ivester received his Ph.D. in Mechanical Engineering from the University of Massachusetts, Amherst in May 1996. Dr. Ivester's primary areas of interest are manufacturing processes, process characterization, optimization, automatic controls, and artificial intelligence.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press

Don Libes (http://www.nist.gov/msidstaff/libes) is a computer scientist at NIST. Don has written 75 computer science papers and articles including three books: *Life With UNIX*, *Obfuscated C and Other Mysteries*, and *Exploring Expect: A Tcl-Based Toolkit for Automating Interactive Programs*.

Simon Szykman is a mechanical engineer in the Manufacturing Systems Integration Division at NIST, where he is currently leading an effort in Intelligent and Distributed Design. He also recently served as the Program Manager for the Advanced Technology Program (ATP) Technologies for the Integration of Manufacturing Applications focused program. Dr. Szykman's research activities have included the use of stochastic optimization methods for product layout, application of optimization to assembly design, and Internet-based computer aided design. His current projects involve the development of knowledge representations and computational tools for modeling of design artifact information to support interoperability in next-generation CAE systems.

https://doi.org/10.1017/S0890060400144038 Published online by Cambridge University Press