The TOVE Project
Towards a Common-Sense Model of the Enterprise

Mark S. Fox

Department of Industrial Engineering
University of Toronto
4 Taddle Creek Road
Toronto, Ontario M5S 1A4

tel: +1-416-978-6823
fax: +1-416-971-1373
email: msf@ie.utoronto.ca

April 27, 1992

Abstract

The goal of the TOVE project is fourfold: 1) to create a shared representation (aka ontology) of the enterprise that each agent in the distributed enterprise can jointly understand and use, 2) define the meaning of each description (aka semantics), 3) implement 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 4) define a symbology for depicting a concept in a graphical context. The model is multi-level spanning conceptual, generic and application layers. The generic and application layers all also stratified and composed of micro theories spanning, for example, activities, time, resources, constraints, etc. at the generic level. Critical to the TOVE effort is enabling the easy instantiation of the model for a particular enterprise TOVE models will be automatically created as a by product of the enterprise design function. TOVE is currently being built to model a computer manufacturer and an aerospace engineering firm.

This research has been supported, in part, by the Natural Science and Engineering Research Council, Digital Equipment Corp., Micro Electronics and Computer Technology Corp., and Spar Aerospace Ltd.

Table of Contents

1. Introduction

Over the last 10 years there has been a shift in how we view the operations of an enterprise. Rather than view the enterprise as being hierarchical in both structure and control, a distributed view where organizational units communicate and cooperate in both problem solving and action has evolved [Fox 81]. Enterprise Integration is concerned with how to improve the performance of distributed organizations and markets. It focuses on the communication of information and the coordination and optimization of enterprise decisions and processes in order to achieve higher levels of productivity, flexibility and quality. To achieve integration it is necessary that units of the enterprise, be they human or machine based, be able to understand each other. Therefore the requirement exists for a language in which enterprise knowledge can be expressed. Minimally the language provides a means of communicating among units, such as design, manufacturing, marketing, field service, etc. Maximally the language provides a means for storing knowledge and employing it within the enterprise, such as in computer-aided design, production control, etc.

We distinguish between a language and a representation. A language is commonly used to refer to means of communication among people in the enterprise. Whereas a representation refers to the means of storing information in a computer (e.g., database). A representation is a set of syntactic conventions that specify the form of the notation used to express descriptions, and a set of semantic conventions that specify how expressions in the notation correspond to things described. With the advent of distributed systems, we are seeing the need for processes (aka agents) to communicate directly with each other. As a result, the representation has become the language of communication. For example, in an object oriented system, we both store and communicate objects without distinction.

The problem that we face today, is that the computer systems to support enterprise functions were independently created, consequently they do not share the same representations. This has led to different representations of the same enterprise knowledge and as a consequence, the inability of these functions to share knowledge. Secondly, these representations are defined without an adequate specification of what the terminology means (aka semantics). This leads to inconsistent interpretations and uses of the knowledge. Lastly, current representations are passive; they do not have the capability to automatically deduce the obvious from what it is representing. For example, if the representation contains a 'works-for' relation and it is explicitly represented that Joe 'works-for' Fred, and that Fred 'works-for' John, then the obvious deduction that Joe 'works-for' John (indirectly) cannot be made within the representation system. The lack of a 'common-sense' deductive capability forces users to spend significant resources on programming each new report or function that is required.

The advent of object-oriented systems does not necessarily resolve any of these concerns. Being object oriented has two interpretations. The more common interpretation is from the programming language perspective: an object is an abstract data type which supports polymorphic invocation of procedures. Consequently the programming paradigm changes from procedure invocation to message sending. The second interpretation is representational. An object represents both classes and instances of things, and they have properties that can be inherited along type hierarchies. Either interpretation does not directly solve the problems that we have raised.

The goal of the TOVE project is fourfold: 1) to create a shared terminology (aka ontology) for the enterprise that each agent can jointly understand and use, 2) define the meaning of each term (aka semantics) in a precise and as unambiguous manner as possible, 3) implement 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 4) define a symbology for depicting a term or the concept constructed thereof in a graphical context.

In the following, we review representation efforts of relevance, describe the TOVE project and discuss measurement criteria and limitations of the approach.

2. Enterprise Modeling Efforts

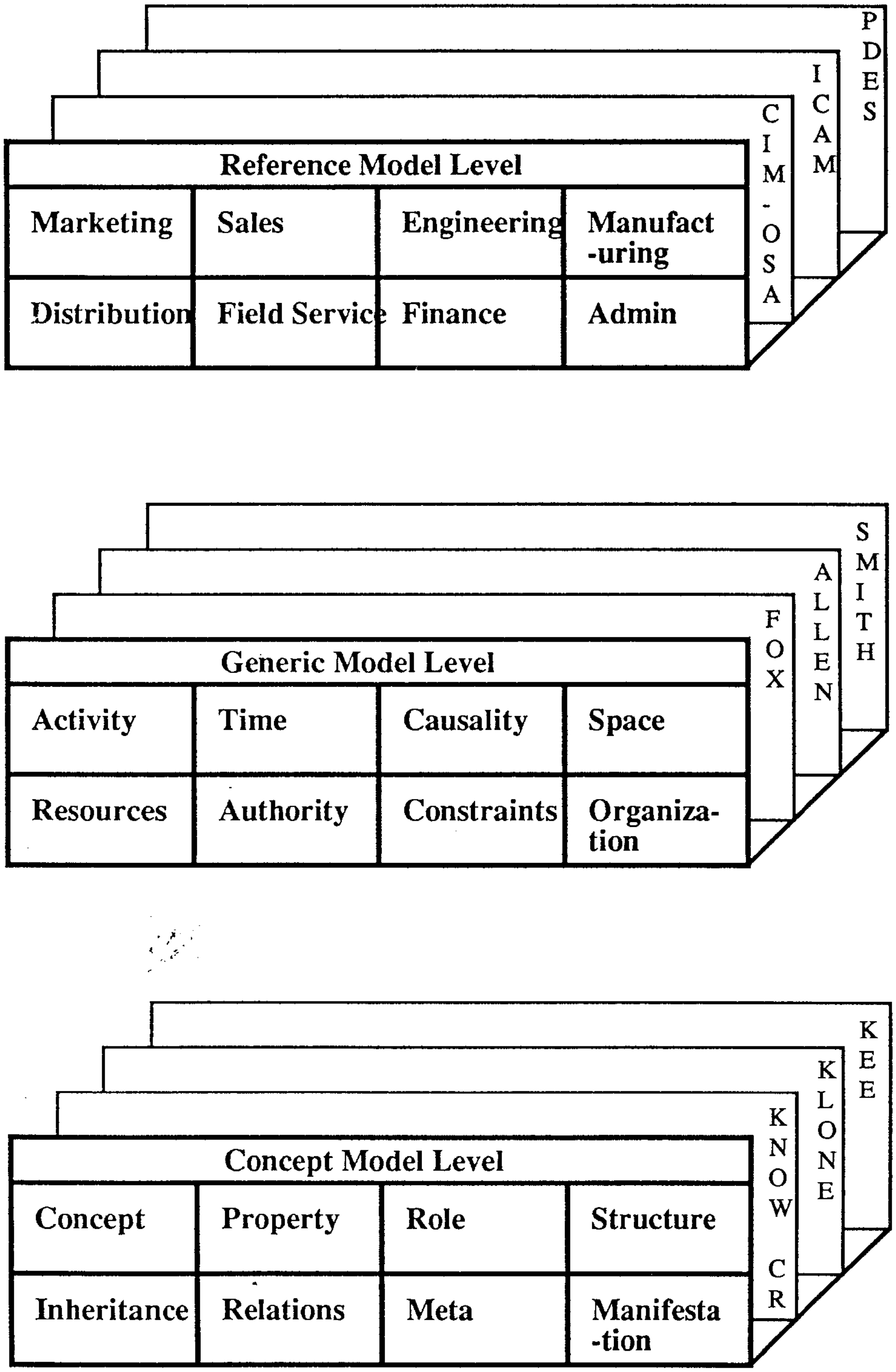
In trying to construct an ontology that spans enterprise knowledge, the first question is where to start. Brachman provides a stratification of representations [Brachman 79]:

- **Implementation:** how to represent nodes and links.
- **Logical:** nodes are predicates and propositions. Links are relations and quantifiers.
- **Conceptual (aka Epistemological):** units, inheritance, intension, extension, knowledge structuring primitives.
- **Generic:** small sets of domain independent elements.
- **Application (aka Lexical):** primitives are application dependent and may change meaning as knowledge grows.

The conceptual level received much attention in the 1970s, with the development of knowledge representation languages such as FRL [Roberts & Goldstein 77], KRL [Bobrow & Winograd 77], SRL [Fox 79], KLONE [Brachman 77] and NETL [Falhman 77]. More recently, there has been a resurgence in interest in conceptual level representations both from a logic perspective, i.e., terminological logics, and a standards perspective. In the 1980s, attention turned to Generic level representations of concepts such as Time [Allen 84], Causality [Rieger & Grinberg 77, Bobrow 85], Activity [Sathi et al. 85], and Constraints [Fox 83, Davis 87]. CYC represents a seminal effort in codifying, extending and integrating generic level concepts [Lenat & Guha 90].

At the application level, various efforts exist in standardizing representations. For example, since the 1960's IBM's COPIC's Manufacturing Resource Planning

Common Sense Model of the Enterprise
Levels of Representation



msf 29 Oct 91

(MRP) system has had a shared database with a single representation of corporate knowledge. In fact, any MRP product contains a standard representation. Recently, several efforts have been underway to create more comprehensive, standard representations of industrial knowledge, including:

CAMI: A US-based non-profit group of industrial organizations for creating manufacturing software and modelling standards.

CIM-OSA: A reference model being developed by the ACIME group of ESPRIT in Europe [Yeomans et al. 85] [ESPRIT-AMICE 90].

ICAM: A project run by the Materials Lab. of the US Air Force [Davis et al. 83] [Martin et al. 83] [Martin & Smith 83, Smith et al. 83].

IWI: A reference model developed at the Institut fur Wirtschaftsinformatik, Universitat des Saarlandes, Germany [Scheer 89].

PDES: Product Data Exchange Standard. Defined by a standards group initially to cover geometric information but then extended to cover additional product data. The model provides a deep view of product descriptions but does not address enterprise modeling.

Though all of these efforts seek to create a sharable representation of enterprise knowledge, there has neither been a well defined set of criteria that these efforts should satisfy, nor has a formal underlying ontology and semantics been created to enable common-sense reasoning. Consequently, their interpretation varies from user to user.

3. The TOVE Project

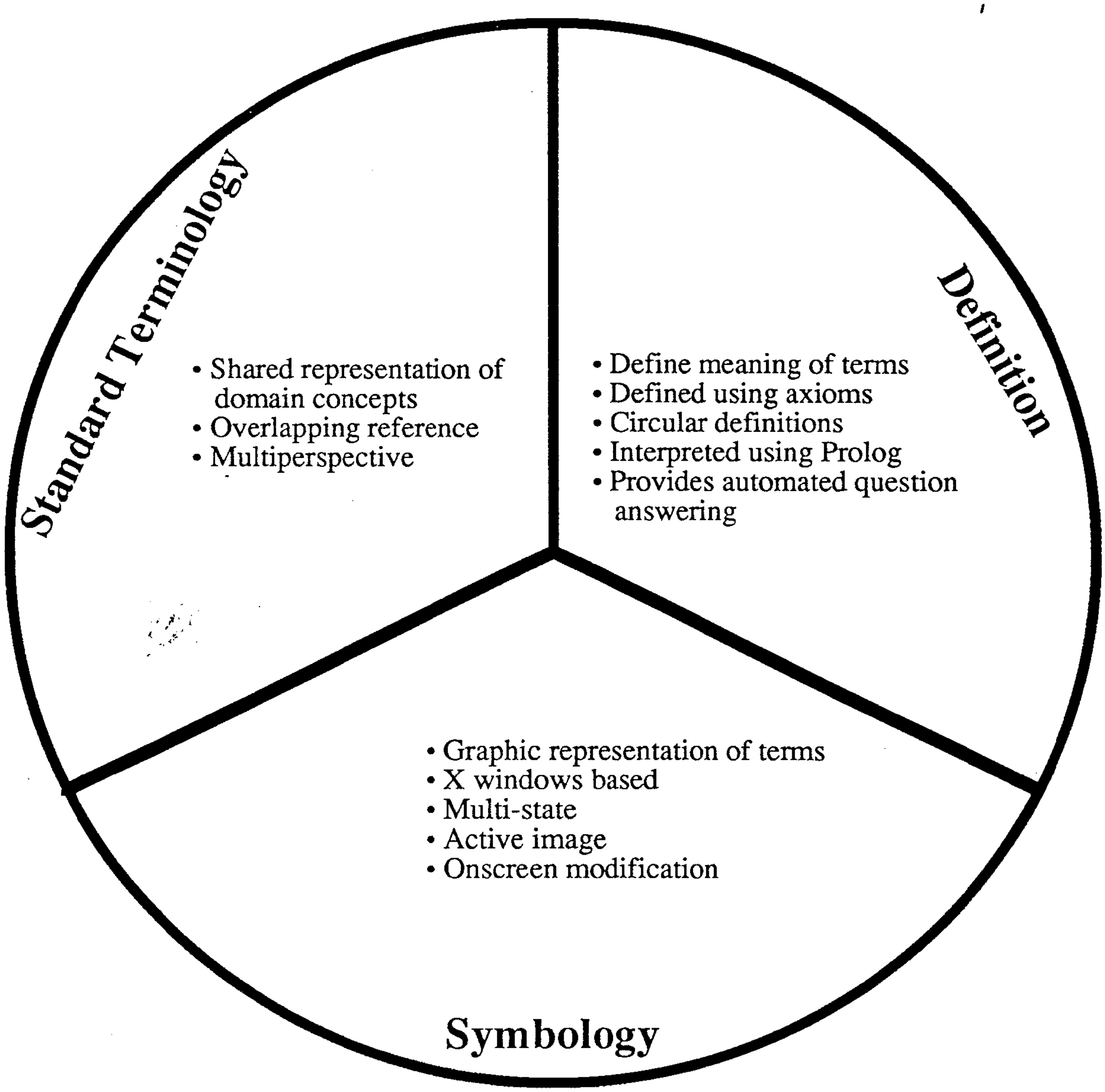
As stated above, the goal of the TOVE project is fourfold: 1) to create a shared terminology (aka ontology) for the enterprise that each agent can jointly understand and use, 2) define the meaning of each term (aka semantics), 3) implement the semantics in a set of axioms that will enable TOVE to automatically deduce the answer to many "common sense" questions about the enterprise, and 4) define a term and concept symbology.

We are approaching the first goal by defining a reference model for the enterprise. A reference model provides a data dictionary of concepts that are common across various enterprises, such a products, materials, personnel, orders, departments, etc. It provides a common model that represents a starting point for the creation of an enterprise specific model. Our reference model will incorporate standard models, where available, e.g., CIM-OSA, IWI, ICAM, CAMI, but deviate from standards where research dictates.

We approach the second goal by defining a generic level representation in which the application representations are defined in terms of. Generic concepts include representations of Time [Allen 84], Causality [Rieger & Grinberg 77, Bobrow 85], Activity [Sathi et al. 85], and Constraints [Fox 83, Davis 87]. The generic level is, in turn, defined in terms of an conceptual level based on the 'terminological logic' of

Common Sense Model of the Enterprise

Representation Components



KLONE [Brachman & Schmolze 85].

We approach the third goal by defining at each level of the representation, generic and application, a set of axioms (aka rules) that define common-sense meanings for the terminology. We view definitions as being mostly circuitous, as opposed to be reducible to a small set of grounded terms. The axioms can be used to deduce the answers to many questions that will be posed by users.

4. Towards A Microtheory for Resources

An example of a generic level representation is a "microtheory" for resources. A microtheory is a locally consistent syntax and semantics for the representation of some portion of knowledge[1].

We view that "being a resource" is not innate property of an object, but is a property that is derived from the role an entity plays with respect to an activity. Consider the role of a steel bar in the activity of machining it into a 3D shape. Properties that derive from an object's role as a resource in this activity may include:

- **Consumption:** A resource is "used" or "used up" by an activity. The latter indicates that the resource, once used, is no longer available in its original form once the activity is completed. In fact, its former self may no longer exist. "Using" a resource indicates the original resource exists after the completion of the activity.
- **Divisibility:** A divisible resource, like water, is still the same resource no matter how you divide it - to some limit. Divisibility can occur along a physical or temporal dimension.
  - **Physical Structure:** Resources may be randomly, physically divisible, such as fluids, or in a structured manner, such as an oven. The nature of the structuring my be imposed by its role.
  - **Temporal Structure:** Resources may be temporally divided either randomly, such as a pizza oven, or in a structured manner, such as a communication line or autoclave. Again it depends on its role.
- **Resource Availability:** The availability of a resource for usage is a characteristic of both consumable and reusable resources. Given a role, a resource may have a maximum capacity.

A set of axioms have been defined that relate and operationalize the meaning of these properties.

[1]Micro-theories have been used extensively in the CYC project at MCC [Lenat & Guha 90].

5. Measurement Criteria

The success of this project can be measured in two ways. The first measure is the extent to which the representation models successfully two or more enterprises. The second approach focuses on the intrinsic characteristics of the representation:

- ***Generality:*** To what degree is the representation shared between diverse activities such as design and troubleshooting, or even design and marketing? What concepts does it span?

- ***Competence:*** How well does it support problem solving? That is, what questions can the representation answer or what tasks can it support?

- ***Efficiency:*** Space and inference. Does the representation support efficient reasoning, or does it require some type of transformation?

- ***Perspicuity:*** Is the representation easily understood by the users? Does the representation "document itself?"

- ***Transformability:*** Can the representation be easily transformed into another more appropriate for a particular decision problem?

- ***Extensibility:*** Is there a core set of ontological primitives that are partitionable or do they overlap in denotation? Can the representation be extended to encompass new concepts?

- ***Granularity:*** Does the representation support reasoning at various levels of abstraction and detail?

- ***Scalability:*** Does the representation scale to support large applications?

- ***Integration:*** Can the representation be used directly or transformed so that its content can be used by existing analysis and support tools of the enterprise?

Satisfaction of these criteria directly affect its acceptability within the enterprise and ultimately its ability to increase the productivity and quality of decisions and actions.

These criteria bring to light a number of important issues and risks. For example, where does the representation end and inference begin? Consider the competence criterion. The obvious way to demonstrate competence is to define a set of questions that can be answered by the representation. If no inference capability is to be assumed, then question answering is strictly reducible to "looking up" an answer that is represented explicitly. In contrast, Artificial Intelligence representations have assumed at least inheritance as a deduction mechanism. In defining a shared representation, a key question then becomes: should we be restricted to just an ontology? Should the ontology assume an inheritance mechanism at the conceptual level, or some type of theorem proving capability as provided, say, in a logic programming language with axioms restricted to Horne clauses (i.e., Prolog)? What is the *deductive capability* that is to be assumed by a reusable representation?

The efficiency criterion is also problematic. Experience has demonstrated clearly that there is more than one way to represent the same knowledge, and they do not have the same complexity when answering a specific class of questions. Consequently, we cannot assume that the representation will partition the space of concepts, but there will exist overlapping representations that are more efficient in answering certain questions. Secondly, the deductive capability provided with the representation affects the store vs compute tradeoff. If the deduction mechanisms are taken advantage of, certain concepts can be computed on demand rather than stored explicitly.

The ability to validate a proposed representation is critical to this effort. The question is: how are the criteria described above operationalized? The *competence* of a representation is concerned with the span of questions that it can answer. We propose that for each category of knowledge within a partition and for each partition, a set of questions be defined that the representation can answer. Given a conceptual level representation and an accompanying theorem prover (perhaps prolog), questions can be posed in the form of queries to be answered by the theorem prover. Given that a theorem prover is the deduction mechanism used to answer questions, the *efficiency* of a representation can be defined by the number of LIPS (Logical Inferences Per Second) required to answer a query. Validating *generality* is more problematic. This can be determined only by a representation's consistent use in a variety of applications. Obviously, at the generic level we strive for wide use across many distinct applications, whereas at the application level, we are striving for wide use within an application.

6. Problems in Usage

The effort in creating an Enterprise model is fraught with problems. The identification of measurement criteria is one step towards being able to compare alternatives. But there are other problems that are not addressed by these criteria. One is the **Correspondence Problem**. What is the relationship among concepts that denote the same thing but have different terminological descriptions? It is common for enterprises, especially those that cross country boundaries to use different names to refer to the same concept. No matter how rationale the idea of renaming them is, organizational barriers impede it.

Another problem is the sheer size of the model. Consider the following basic relations, and objects in their range, defined for the "part" concept in the ICAM model from the design perspective [Martin et al. 83] [Martin & Smith 83]:

- **IS CHANGED BY:** Part Change (105) (also shown as "is modified by")
- **APPEARS AS:** Next Assembly usage item (119) (also shown as "is referenced as").
- **HAS:** Replacement part (143).
- **HAS SUBTYPE (IS):** Parts list item (118), Replacement part (143).
- **IS USED AS:** Next Assembly Usage (40), Advance material notice item part (144), Configuration list item (170).

- **IS TOTALLY DEFINED BY:** Drawing (1).
- **IS LISTED BY (LISTS):** Configuration list (84).
- **IS USED IN:** Effectivity (125).
- **IS FRABRICATED FROM:** Authorized material (145).

and from a manufacturing perspective:

- **HAS:** N.C. Program (318), Material issue (89), Component part (299), Alternative part (301), Part/process specification use (255), Material receipt (87), Work package (380), Part tool requirement (340), Part requirement for material (397), Standard routing use (254), Image part (300), Part drawing (181).
- **IS ASSIGNED TO (HAS ASSIGNED TO IT):** Index (351).
- **IS DEFINED BY (DEFINES):** Released engineering drawing (12).
- **IS SUBJECT OF:** Quote request (90), Supplier quote (91).
- **IS TRANSPORTED BY:** Approved part carrier (180).
- **IS RECEIVED AS:** Supplier del lot (309).
- **APPEARS AS:** Part lot (93), Ordered part (188), Serialized part instance (147), Scheduled part (409), Requested purchase part (175).
- **CONFORMS TO:** Part specification (120).
- **IS INVERSE:** Component part (299), Alternate part (301), Section (363), End item (5), Configured item (367), Image part (300).
- **IS USED AS:** Component part callout (230), Process plan material callout (74).
- **IS SUPPLIED BY:** Approved part source (177).
- **MANUFACTURE IS DESCRIBED BY:** Process plan (415).
- **SATIFIES:** End item requirement for part (227).
- **IS REQUESTED BY:** Manufacturing request (88).
- **IS STORED AT:** Stock location use for part (227).
- **IS SPECIFIED BY:** BOM Item (68).

There are a number of problems with this model. First, the meanings of the relations are ambiguous and at best provided in a descriptive form. Secondly, there is no way to know whether this is the "right" way of representing this information. Third, this is only the tip of the iceberg. If one were to develop a complete model for an enterprise, its sheer size would be beyond the abilities of any database manager or knowledge engineer to understand and use effectively. Consequently, the creation of a enterprise model for a particular firm may have to be performed in another way.

Our recommendation is that the creation of a firm's enterprise model be a byproduct of the the enterprise design function. Our view is similar to that of the IDEF family of modeling languages in that it is the design of the firm's activities that entail a subset of enterprise modeling classes to be instantiating. The result of specifying an enterprise's goals and activities should be an enterprise model. But in order to successfuly generate a model, the activity modeling methods most be more explicit in the specification of goals, activities, constraints, resources, etc. than is currently found in IDEF-like modeling tools.

7. TOVE Testbed

TOVE is not only a research project but an environment in which to perform research. The TOronto Virtual Enterprise (TOVE) is a virtual company whose purpose is to provide a testbed for research into enterprise integration. TOVE grew out of need to provide a single testbed that would integrate our research efforts. Our short term goal for TOVE is to define a company, existing solely in the computer, to support the exploration of issues in planning, and scheduling with fully specified models for both flowshop and jobshop experiments. Consequently, the criteria for selecting a product that TOVE produces includes:

- It would provide a testbed for primarily mechanical design, with the opportunity for electrical and electronic design. A domain that involves description and manipulation of 3D objects which have interesting but not too detailed design features.
- It could be designed to be as simple or complex as desired.
- Components would have to be fabricated and assembled so that planning and scheduling research could explore both.
- Components could be made out of a variety of materials, both mundane and exotic.
- A variety of resources and processes which provide complex challenges for process planning, facility layout, and scheduling systems.
- Components could actually be fabricated at CMU or purchased externally.
- Students and faculty at CMU would want to purchase it.

Desk lamps were selected with these criteria in mind. Lamp components fit the design criterion quite well, as many are relatively simple, but all have at least a few interesting and unique features. For example, some arm components are straightforward hollow cylinders, while some base and head components are irregular polygons in 3D. With respect to the materials criterion, lamp components can be metal, hard plastic, soft plastic, wire, and foam. Some components can actually be either metal or plastic. With respect to the material and process variety criteria, lamp manufacturing requires purchasing, fabrication, assembly, subcontracting, non-destructive testing, packing, and distribution, as well as front end marketing and sales operations. The resources for these processes are large in number and type, as well as diverse in their operational and maintenance needs. Parts can be produced either in batches or on an individual basis. Major lamp components are heads, arms, and bases. Three styles of each are produced, with a standard interface between base/arm and arm/head components. The parts mix is achieved with a mix-n-match of these major components.

An earlier version of TOVE, called CARMEMCO, was developed in LISP and Knowledge Craft^{R} at Carnegie Mellon University by Lin Chase. We have adapted this to a C++ environment using the ROCK^{TM} knowledge representation tool from Carnegie Group. TOVE operates "virtually" by means of knowledge-based simulation [Fox et al. 89]. Future versions of TOVE will extend it to be multi-plant and multi-region situations.

8. Conclusion

In conclusion, the TOVE project's goals are 1) to create a shared terminology (aka ontology) of the enterprise that each agent can jointly understand and use, 2) define the meaning of each term (aka semantics), 3) implement the semantics as a set of axioms that will enable TOVE to automatically deduce the answer to many "common sense" questions about the enterprise, and 4) define a symbology for depicting terms and concepts in a graphical context. We are approaching these goals by defining a three level representation: application, generic and conceptual. Each level will have a well-defined terminology which will be defined in terms of lower level terms. Each term and each level will have an axiomatic definition of it terms, comprising a micro theory for a subset of terms at that level, and enabling the deduction of answers to common-sense questions. The creation of a TOVE model will be the by product of the activity model of the enterprise. TOVE is currently being built to model two enterprises:a computer manufacturer and an aerospace engineering firm.

9. Ackknowledgements

This research is supported in part by an NSERC Industrial Research Chair in Enterprise Integration, Carnegie Group Inc., Digital Equipment Corp., Micro Electronics and Computer Research Corp., Quintus Corp., and Spar Aerospace Ltd.

Lin Chase designed and implemented the precurser to TOVE, called CARMEMCO. Donald Kosy and Marty Tenenbaum have contributed both ideas and inspiration to our modelling effort. Section 5 first appeared as part of a DARPA Knowledge Representation workshop white paper by Tenenbaum and myself. Most of section 7 appeared in a CARMEMCO description by Lin Chase.

References

[Allen 84] Allen, J.F.
Towards a General Theory of Action and Time.
Artificial Intelligence 23(2):123-154, 1984.

[Bobrow 85] Bobrow, D.G.
Qualitative Reasoning About Physical Systems.
MIT Press, 1985.

[Bobrow & Winograd 77]
Bobrow, D., and Winograd, T.
KRL: Knowledge Representation Language.
Cognitive Science 1(1), 1977.

[Brachman 77] Brachman, R.J.
A Structural Paradigm for Representing Knowledge.
PhD thesis, Harvard University, 1977.

[Brachman 79] Brachman, R.J.
On the Epistemological Status of Semantic Networks.
Associative Networks: Representation and Use of Knowledge by Computers.
In Findler, N.V.,
Academic Press, 1979, pages 3-50.

[Brachman & Schmolze 85]
Brachman, R.J., and Schmolze, J.G.
An Overview of the KL-ONE Knowledge Representation Systems.
Cognitive Science 9(2), 1985.

[Davis 87] Davis, E.
Constraint Propagation with Interval Labels.
Artificial Intelligence 32:281-331, 1987.

[Davis et al. 83] Davis, B.R., Smith, S., Davies, M., and St. John, W.
Integrated Computer-aided Manufacturing (ICAM) Architecture Part III / Volume III: Composite Function Model of "Design Product" (DES0).
Technical Report AFWAL-TR-82-4063 Volume III, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433, 1983.

[ESPRIT-AMICE 90]
ESPRIT-AMICE.
CIM-OSA - A Vendor Independent CIM Architecture.
In *Proceedings of CINCOM 90,* pages 177-196. National Institute for Standards and Technology, 1990.

[Falhman 77] Fahlman, S.E.
A System for Representing and Using Real-World Knowledge.
PhD thesis, Massachusetts Institute of Technology, 1977.

[Fox 79] Fox, M.S.
On Inheritance in Knowledge Representation.
In *Proceedings of the International Joint Conference on Artificial Intelligence.* 95 First St., Los Altos, CA 94022, 1979.

[Fox 81] Fox, M.S.
An Organizational View of Distributed Systems.
IEEE Transactions on Systems, Man, and Cybernetics SMC-11(1):70-80, 1981.

[Fox 83] Fox, M.S.
Constraint-Directed Search: A Case Study of Job-Shop Scheduling.
PhD thesis, Carnegie Mellon University, 1983.
CMU-RI-TR-85-7, Intelligent Systems Laboratory, The Robotics Institute, Pittsburgh,PA.

[Fox et al. 89] Fox, M.S., Reddy, Y.V., Husain, N., McRoberts, M.
Knowledge Based Simulation: An Artificial Intelligence Approach to System Modeling and Automating the Simulation Life Cycle.
Artificial Intelligence, Simulation and Modeling.
In Widman, L.E.,
John Wiley & Sons, 1989.

[Lenat & Guha 90]
Lenat, D., and Guha, R.V.
Building Large Knowledge Based Systems: Representation and Inference in the CYC Project.
Addison Wesley Pub. Co., 1990.

[Martin & Smith 83]
Martin, C., and Smith, S.
Integrated Computer-aided Manufacturing (ICAM) Architecture Part III / Volume IV: Composite Information Model of "Design Product" (DES1).
Technical Report AFWAL-TR-82-4063 Volume IV, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433, 1983.

[Martin et al. 83] Martin, C., Nowlin, A., St. John, W., Smith, S., Ruegsegger, T., and Small, A.
Integrated Computer-aided Manufacturing (ICAM) Architecture Part III / Volume VI: Composite Information Model of "Manufacture Product" (MFG1).
Technical Report AFWAL-TR-82-4063 Voluem VI, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433, 1983.

[Rieger & Grinberg 77]
Rieger, C., and Grinberg, M.
The Causal Representation and Simulation of Physical Mechanisms.
Technical Report TR-495, Dept. of Computer Science, University of Maryland, 1977.

[Roberts & Goldstein 77]
Roberts, R.B., and Goldstein, I.P.
The FRL Manual.
Technical Report MIT AI Lab Memo 409, Massachusetts Institute of Technology, 1977.

[Sathi et al. 85] Sathi, A., Fox, M.S., and Greenberg, M.
Representation of Activity Knowledge for Project Management.
IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-7(5):531-552, September, 1985.

[Scheer 89] Scheer, A-W.
Enterprise-Wide Data Modelling: Information Systems in Industry.
Springer-Verlag, 1989.

[Smith et al. 83] Smith, S., Ruegsegger, T., and St. John, W.
Integrated Computer-aided Manufacturing (ICAM) Architecture Part III / Volume V: Composite Function Model of "Manufacture Product" (MFG0).
Technical Report AFWAL-TR-82-4063 Volume V, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433, 1983.

[Yeomans et al. 85]
Yeomans, R.W., Choudry, A., and ten Hagen, P.J.W.
Design Rules for a CIM System.
Elsevier Science Publishing Company, 1985.