On Ontologies And Enterprise Modelling

Mark S. Fox & Michael Grüninger

Enterprise Integration Laboratory
Dept. of Mechanical and Industrial Engineering, University of Toronto
4 Taddle Creek Road, Toronto, Ontario M5S 3G9 CANADA
tel: +1-416-978-6823; fax: +1-416-971-2479;
email: msf@ie.utoronto.ca, mudcat@ie.utoronto.ca

1.0 Introduction

An Enterprise Model is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints of a business, government, or other enterprise. It can be both descriptive and definitional - spanning what is and what should be. The role of an enterprise model is to achieve model-driven enterprise design, analysis and operation.

From a design perspective, an enterprise model should provide the language used to explicitly define an enterprise. We need to be able to explore alternative models in the design of enterprises spanning organisation structure and behaviour. In order to reason about alternative designs for enterprises, we need to reason about different possible sets of constraints for enterprises within the model. We need to ask the questions -- can a process be performed in a different way, or can we achieve some goal in a different way? Can we relax the constraints in the enterprise such that we can improve performance or achieve new goals?

We also need to be able to determine the impact of changes on all parts of the enterprise. For example, if we relax one of the policies, how will this affect the quality of products or services provided by the enterprise? If we purchase a new kind of machine, how will this affect the activities that are performed? Will we need to retrain people in the enterprise to give them the skills to use the machine? If we change the activities that are performed, how will this change resource consumption?

From an operations perspective, the enterprise model must be able to represent what is planned, what might happen, and what has happened. It must supply the information and knowledge necessary to support the operations of the enterprise, whether they be performed manually or by machine. It must be able to provide answers to questions commonly asked in the performance of tasks.

Its clear from these requirements that we need more than a simple data modelling. What is needed is the ability to deduce what is implied by the model.

In this paper we introduce Ontologies as a basis for modelling enterprises. An ontology "consists of a representational vocabulary with precise definitions of the meanings of the terms of this vocabulary plus a set of formal axioms that constrain the interpretation and well-formed use of these terms." [Campbell & Shapiro 95]. Axioms provide the basis of an ontology's deductive capability. We begin by introducing the concept of a Generic Enterprise Model (GEM). We then extend the concept of a GEM to a Deductive Enterprise Model (DEM) and then briefly review research to date.

2.0 Generic Enterprise Models (GEM)

Over the last 30 years, the role of enterprise models in the design and operation of enterprises has reached the point that few organisations of significant size can operate with out them. For example, Manufacturing Requirements Planning systems have at their core a data model of the organisation spanning resources, activities, and products. They use this model to plan and control operations. Today, MRP systems have evolved into Enterprise Requirements Planning systems where the enterprise model is viewed as a major component[1]. Similarly Business Process Re-engineering tools, such as FirstStep from Interfacing Technologies, Bonapart from UBIS GmbH, and ReThink from Gensym, have at their core an Enterprise Model[2]. It would not be overly general to say that most information systems in use within an enterprise incorporate a model of some aspect of its structure, operations and/or knowledge.

The problem that we face today, is that the legacy systems that support enterprise functions were created independently, consequently do not share the same enterprise models. We call this the *Correspondence Problem*. Though each enterprise model may represent the same concept, e.g., activity, they will have a different name, e.g., operation vs. task. Consequently, communication among functions is not possible without translation. No matter how rational the idea of renaming them is, organisational barriers impede it. Secondly, these representations lack an adequate specification of what the objects (terminology) mean (aka semantics). This leads to inconsistent interpretations and uses of the knowledge. Lastly, the cost of designing, building and maintaining a model of the enterprise is large. Each tends to be unique to the enterprise; objects are enterprise specific.

As a solution to this problem, there has been an increasing interest in Generic Enterprise Models (GEM). A GEM is an object library that defines the classes of objects that are generic across a type of enterprise, such as manufacturing or banking, and can be employed (aka instantiation) in defining a specific enterprise. A GEM is composed of the following:

- A set of object classes structured as a taxonomy, i.e., each object is linked to one or more other objects by a sub-class/super-class relationship plus a definition of how a class refines its super-class.
- For each object class a set of relations are defined linking it to other object classes, plus a definition of the intended meaning of each relation.
- For each object class a set of attributes plus a definition of the intended meaning of each attribute.

The benefits of employing a GEM at the outset when creating an enterprise model are many:

- Pre-defined Object Library - most database engineers often start from scratch when creating an enterprise model. Defining the “right” set of object classes is a daunting and time consuming task. A GEM provides the object classes, allowing the engineer to quickly move on to model instantiation.

1. In fact, the importance of enterprise modelling was recognised by industry’s decision to create a parallel organisation to the Object Management Group (OMG: www.omg.org) to focus on the definition of standard business objects.
2. See [Spurr et al. 94] for a review of Reengineering tools.

- Path for Growth - many enterprise modellers do not know what they have “left o ut” until it is too late. By incorporating a GEM, many of the concepts that they may not have anticipated they would need are already there; the modeller has benefited from the experience of others.
- Shared Conceptualization - By adopting a GEM, other parts of the organisation stand a greater chance of understanding what is represented in the enterprise model.

Ultimately, these benefits affect the bottom line. Both time and costs are reduced.

3.0 Common Sense Enterprise Models and Ontologies

The usefulness of an instantiated GEM is determined by the functions it can support, e.g., scheduling, forecasting, accounting. Since the interface to an enterprise model is through the query language provided by the underlying database, the functions a GEM can support is determined by the categories of queries the GEM can provide answers to (if properly instantiated). But the queries that a GEM can answer are not just determined by the object library and its instantiations, but by additional processing that may be provided.

Where does the GEM end and inference begin? If no inference capability is to be assumed, then question answering is strictly reducible to “looking up” an answer that is represented explicitly in the model. In contrast, current object models have assumed at least inheritance as a deduction mechanism; answers can be provided that assume properties of the class apply to an instance. In defining an enterprise model, a key question then becomes: should we be restricted to just an object library? Should the objects assume an inheritance mechanism, or some type of theorem proving capability, as provided, say, in a logic programming language with axioms restricted to Horne clauses (i.e., Prolog)? In other words, what is the *deductive capability* that is to be assumed by a GEM?

We introduce three types of queries: Factual, Expert and Common-Sense. Consider a relational database system. Such databases support factual queries by the direct retrieval of information represented explicitly in the model (i.e., surface level processing). Consider a model with an SQL interface. Information is explicitly represented if it can be retrieved using a simple SELECT. For example, if the model contains a ‘works-for’ relation and it is explicitly represented that Joe ‘works-for’ Fred, then the database can return the answer “Fred” in response to a query of “who does Joe works-for”[1].

Expert queries require that the information system have extensive knowledge and reasoning capabilities (i.e., deep level processing). Expert systems [Fox 90] provide deep level processing. By deep level we mean that a significant amount of search, i.e., deductions, have to be performed in order to provide a response to a query. In order to answer a query regarding the cause of a machine malfunction, the expert system might have to reason about the structure and behaviour of the machine. It must have a detailed model of the domain and it may be unique to the specific enterprise. Such systems tend to be costly to build and maintain and are narrow in scope.

---

1. Given a relation works-for(Supervisor, Supervisee), then the SQL query would be: (SELECT Supervisor FROM works-for WHERE Supervisee = Joe).

Common-Sense queries require that the information system be able to deduce answers to questions that one would normally assume can be answered if one has a "commons-sense" understanding of the enterprise. This often represents knowledge about the enterprise acquired over a relatively short period of time, e.g., 3-9 months, and does not denote knowledge of an expert nature. For example, if the model 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) could be deduced using "common-sense"[1]; that is, we have a common-sense understanding of the meaning of the relation "works-for". It could be argued that the majority of queries posed to a database are in this third category: common-sense. That if GEM's were designed to support common-sense queries, a significant portion of the MIS backlog could be done away with.

Common-sense query processing assumes a third level of processing that we shall refer to as shallow level processing. By shallow level, we mean retrieval that requires a small number of deductions, i.e., 1-1000, in order to answer the query. In order for an enterprise model to support common-sense query processing, it must provide a set of rules of deduction, aka axioms. For the 'works-for' example, we would require an axiom stating that 'works-for' is transitive:

x works-for y AND y works-for z IMPLIES x works-for z. (EQ 1)

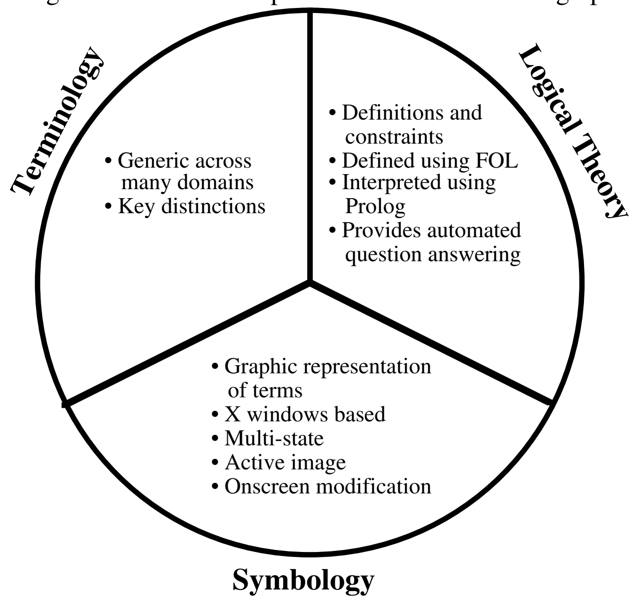
We distinguish between an enterprise model that includes axioms that support deduction, versus a model without axioms where deductions are specified by the query. In the former case, the model would be able to deduce that Joe works-for John in response to a query asking "who does Joe work for?" In the latter case, the user would have to specify a complex query which would include as many joins as necessary to travel along the 'works-for' relation. Since the user does not know at the outset the depth of the 'works-for' path, they may not get the information they were looking for. We call a GEM that includes axioms and a deduction engine (i.e., theorem prover or a deductive database) a Deductive Enterprise Model (DEM). We say a DEM possesses Common-Sense if its axioms define the meaning of the relations and attributes in the object library.

It is our belief that the design and construciton of a DEM can best be addressed by taking a more formal approach to enterprise modelling. Over the last five years the field of Ontological Engineering has emerged. (See the *Proceedings of the AAAI Spring Symposium on Ontological Engineering*, AAAI Press, 1997.) An ontology is a formal description of entities and their properties; it forms a shared terminology for the objects of interest in the domain, along with definitions for the meaning of each of the terms.

The following figure depicts the basic components of an ontology: 1) it provides a shared terminology for the enterprise that every application can jointly understand and use, 2) it defines the meaning (semantics) Of each term in a precise and as unambiguous manner as possible using First Order Logic, 3) it implements the seman-

1. A relational database cannot support common-sense queries directly. Instead, the commons-sense query would have to specified as a series of factual queries containing one or more JOINs combined with SELECTs. This is equivalent to performing deduction. The lack of a `common-sense' deductive capability forces users to spend significant resources on programming each new report or function that is required.

tics in a set of Prolog axioms that enable model to automatically deduce the answer to many “common sense” questions about the enterprise, and 4) it defines a symbology for depicting a term or the concept constructed thereof in a graphical context.



4.0 Evaluating Ontologies

How can you determine which ontologies (DEM[1]s) are right for our task? Following are what we believe should be the characteristics with which to evaluate an ontolgy:

Functional Completeness: Can the ontology represent the information necessary for a function to perform its task.

Generality: To what degree is the ontology shared between diverse activities such as engineering design and manufacturing, or design and marketing? Is the ontology specific to a sector, such as manufacturing, or applicable to other sectors, such as retailing, finance, etc.

Efficiency: Does the ongology support efficient reasoning, i.e., space and time, or does it require some type of transformation?

Perspicuity: Is the ontology easily understood by the users so that it can be consistently applied and interpreted across the enterprise? Does the representation "document itself?”

Precision/Granularity: Is there a core set of ontological primitives that are partitionable or do they overlap in meaning? Does the representation support reasoning at various levels of abstraction and detail?

Minimality: Does the ontology should contain the minimum number of objects

1. Since a GEM is a subset of a DEM - it does not contain the axioms - we will only refer to DEMs from here on in.

(i.e., terms or vocabulary) necessary [Gruber 93].

The problem is: how are these criteria operationalized?

We introduce the concept of a ontology's *competence* [Gruninger & Fox 94]. Given a properly instantiated model of an enterprise and an accompanying theorem prover (perhaps Prolog or a deductive database), the *competence* of a ontology is the set of queries that it can answer. Ideally, the competency questions should be defined in a stratified manner, with higher level questions requiring the solution of lower level questions. Another view of competency is that it evaluates the expressiveness of the ontology that is required to represent the competency questions and to characterize their solutions[1].

Using the concept of competency, we can determine how well a particular ontology satisfies them as follows.

The *Functional Completeness* of a ontology is determined by its competency, i.e., the set of queries it can answer with a properly instantiated model. Given a particular function (application), its enterprise modelling needs can be specified as a set of queries. If these queries can be "reduced to"[2] the set of competency questions specified for the chosen ontology, then the ontology is sufficient to meet the modelling needs of the application.

The *Generality* of a ontology can be determined by evaluating whether the union of queries from a broad set of functions, perhaps drawn from different sectors, are reducible to an ontology's competency.

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. But experience has demonstrated that there is more than one way to represent the same knowledge, and each representation does not have the same complexity when answering a specific class of questions. Furthermore, the deductive capability provided with the ontology affects the store vs. compute trade-off. If the deduction mechanisms are taken advantage of, certain concepts can be computed on demand rather than stored explicitly. By computing the average complexity of the competency questions of the ontology, we can estimate its efficiency.

The *perspicuity* of a ontology is enhanced by its axiomatisation. That is, by providing formal definitions of objects and their relations and attributes, we make it possible for users to understand their intended meaning. Though this does not guarantee that programs that access a ontology will interpret the results correctly.

The *precision* of a ontology refers to what extent the definitions of concepts are *distinct* [Sowa 95]. Given that we have formal definitions, we can determine whether a concept subsumes another, or what concepts lie at their intersection or union (generalization). *Granularity* refers to the capability of representing concepts at differing levels of abstraction. Whether a ontology is precise enough or allows for the capability of abstraction should be determined from their competency questions. Conversely, an application specifies its precision and granularity requirements in the

---

1. These competency questions do not generate ontological commitments; rather, they are used to evaluate the ontological commitments that have been made.

2. By reducible, we mean the questions can be re-written using the objects provided by the chosen ontology.

form of queries. If these queries can be reduced to the ontology's competency questions, then the ontology is sufficient.

Lastly, the *minimality* of a ontology can be determined, using the axioms, by proving that for every object in the ontology there is no other object that is logically equivalent.

5.0 Ontologies for Enterprise Modelling

The need for Generic Enterprise Models (GEM) became obvious with the availability of MRP systems, manufacturing simulations, financial analysis systems, etc. The reasons were two fold. First, the modules within a system required a common representation of the enterprise. For example, in an MRP system, an operation had to be represented in one way to support the scheduling, general ledger, and purchasing functions. Secondly, in order to train users in how to structure the information about their enterprise, a single modelling language had to be taught. Though generic in design, these GEMs did not result in industry standards.

Attempts to create industry-wide standards began to appear with the US Air Force's ICAM effort in the early 80s. A GEM for the aerospace industry was developed [Martin & Smith 83]. At IWI, Scheer [89] developed a GEM for manufacturing. Another major manufacturing GEM creation effort is the ESPRIT program's CIM-OSA project [AMICE 93]. Related efforts such as PERA and the GRAI are described in [Bernus et al. 96]. The US Department of Defense has also embarked upon the creation of a DOD-wide GEM [DOD 93].

The development of ontologies for Enterprise Models (DEMs) is more recent[1]. There are a few projects whose scope of modelling is rather broad, including the CYC project at MCC [Lenat & Guha 90], the TOVE project at the University of Toronto [Fox et al. 93], and the Enterprise project at the University of Edinburgh [Uschold et al. 97].

What is modelled of an enterprise can be divided into categories. The following identifies a subset of these categories and ontologies being developed for them.

- **Process and Activities:** including the representation of state, time, and causality [Gruninger & Fox 94] [Gruninger & Pinto 95] [Fillion et al. 95] [Menzel & Mayer 96] [Schlenoff et al. 96] [Vernadat 96]. Note that this area has received the greatest attention from the Artificial Intelligence Knowledge Representation and Planning communities.
- **Resources & Inventory**: General representation of resources, inventory, location, etc. [Fadel et al. 94].
- **Organization Structure:** Representation of positions, roles, departments, processes, goals, constraints, etc. [Lee 88] [Fox et al. 95] [Yu et al. 96].
- **Product Structure and Requirements:** [Borgo et al. 96] [Lin et al. 96] [Liebig & Roesner 96].
- **Quality**: Basic representations of quality in support of ISO9000, QFD, etc.[Kim & Fox 95].
- **Cost**: Representation of resource costs, activity costs, Activity-based costing, etc. [Tham et al. 94] [Nado et al. 96].

---

1. Methods for engineering ontologies are described in: [Gruber 93] [Goldstein & Esterline 95] [Grüninger & Fox 95].

What is obvious from the list above is that the field of ontology engineering is emerging. Through much work has been done in creating GEMs, much less has been done to extend them to include a clearly defined semantics.

6.0 Conclusions

The drive for more agile enterprises requires a degree of integration that is not possible without the use of a sophisticated information infrastructure. At the core of this infrastructure lies an enterprise model. Efforts are underway to create Generic Enterprise Models (GEMs), whose use significantly reduces the time to design and implement enterprise models. More recently, Deductive Enterprise Models (DEMs) have been developed that play a more active role in the support of enterprise operations by deducing answers to commonly asked questions. The recent development of Ontology Engineering is providing a formal basis for the development of DEMs.

7.0 Acknowledgements

This research is supported, in part, by the Natural Science and Engineering Research Council of Canada, Carnegie Group Inc., BHP Ltd., Digital Equipment Corp., Micro Electronics and Computer Research Center, Numetrix Ltd., Quintus Corp., Spar Aerospace and Toyo Engineering.

This is a shorten and modified version of Fox, M.S., Grüninger, M., (1997), "Enterprise Modelling", *AI Magazine*, to appear.

8.0 References

[AMICE 93] AMICE, (1993), *CIM-OSA Open System Architecture for CIM, 2nd Revised and Extended Edtion*. Berlin, Germany: Springer-Verlag.

[Bernus et al. 96] Bernus, P., Nemes, L., Williams,T.J., (1996), *Architectures for Enterprise Integration,* Chapman and Hall, London.

[Borgo et al. 96] Borgo, S., Guarino, N., Masolo, C., (1996), "Stratified Ontologies: The Case of Physical Objects", *Proceedings of the Workshop on Ontology Engineering*, ECAI96. http://www.ladseb.pd.cnr.it/infor/Ontology/BaselPapers/Grabowski_ps.Z

[Campbell & Shapiro 95] Campbell, A.E., and Schapiro, S.C., (1995), "Ontologic Mediation: An Overview", *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knoweldge Sharing*, Menlo Park CA, USA: AAAI Press.

[DoD 93] The Office of the Director of Defense Information (no author), (1993), "The DoD Enterprise Model", Office of the Secretary of Defense, Washington D.C., USA.

[Fadel et al. 94] Fadel, F., Fox, M.S., and Gruninger, M., (1994), "A resource ontology for enterprise modelling", *Third Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*, West Virginia University, pp. 117-128.

[Fillion et al. 95] Fillion, F., Menzel, C., Blinn, T., Mayer, R., (1995), "An Ontology-Based Environment for Enterprise Model Integration", *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, AAAI Press*.

[Fox 90] Fox, M.S., (1990), "Aritificial Intelligence and Expert Systems: Myths, Legends and Facts”, *IEEE Expert*.

[Fox et al. 93] Fox, M., Chionglo, J.F., and Fadel, F.G., (1993), “A Common Sense Model of the Enterprise”, *Proceedings of the 2nd Industrial Engineering Research Conference*, pp. 425-429, Norcross GA, USA: Institute for Industrial Engineers.

[Fox et al. 95] Fox, M.S., Barbuceanu, M., Gruninger, M., (1995), “An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour”, *Computers in Industry*, Vol. 29, pp. 123-134.

[Goldstein & Esterline 95] Goldstein, D., and Esterline, A., (1995), “Methods for Building Sharable Ontologies”, *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knoweldge Sharing*, Menlo Park CA, USA: AAAI Press.

[Gruber 93] Gruber, T.R., (1993), “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”, Technical Report, Knowledge Systems Laboratory, Stanford University.

[Gruninger & Fox 94] Gruninger, M., and Fox, M.S., (1994), “The Role of Competency Questions in Enterprise Engineering”*, Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice,* Trondheim, Norway. http://www.ie.utoronto.ca/EIL/public/competency.ps

[Grüninger & Fox 95] Grüninger, M., and Fox, M.S., (1995), “Methodolgy for the Design and Evaluation of Ontologies”, *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knoweldge Sharing*, Menlo Park CA, USA: AAAI Press. http://www.ie.utoronto.ca/EIL/public/org.ps

[Gruninger & Pinto 95] Gruninger, M., and Pinto, J.A., (1995), "A Theory of Complex Actions for Enterprise Modelling", Working Notes AA AI Spring Symposium Series 1995: Extending Theories of Action: Formal Theory and Practical Applications, Stanford. http://www.ie.utoronto.ca/EIL/public/complex.ps

[Kim & Fox 95] Kim, H. and Fox, M.S., (1995), “An Ontology of Quality for Enterprise Modelling,” *Fourth Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*, West Virginia University.

[Lee 88] Lee, R. M., (1988), “Bureaucracies as deontic systems”, *ACM Transactions on Office Information Systems*, Vol. 6, No. 2, pp. 87-108.

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

[Liebig & Roesner] Liebig, T., Roesner, D., (1996), “Modelling of reusable product knowledge in terminological logics - a case study”, *Proceedings of the Workshop on Product Knowledge Sharing for Integrated Enterprises*. First International Conference on Practical Aspects of Knowledge Management, Basel Switzerland. http://www.ladseb.pd.cnr.it/infor/Ontology/BaselPapers/Liebig_ps.Z

[Lin et al. 96] Lin, J., Fox, M.S., and Bilgic, T., (1996), “A Requirements Ontology for Concurrent Engineering”, *Concurrent Engineering: Research and Applications*, Vol. 4, No. 4, pp. 279-291.

[Martin & Smith 83] Martin, C., and Smith, S., (1983), *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,.

[Menzel & Mayer 96] Menzel, C., Mayer, R.J., (1996), “Situations and Processes”, *Concurrent Engineering: Research and Applications*, Vol. 4, No. 3, pp. 229-246.

[Nado et al. 96] Nado, R., Chams, M.M., Delisio, J., Hamscher, W., (1996), “Comet: An Application of Model-Based Reasoning to Accounting Systems”, In *Proceedings of the Eighth Innovative Applications of Artificial Intelligence Conference*, AAAI Press

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

[Schlenoff et al. 96] Schlenoff, C., Knutilla, A., Ray, S., (1996), “Unified Process Specification Language: Requirements for Modeling Process”, NIST Interagency Report 5910, Gaithersburg, MD.

[Sowa 95] Sowa, J.F., (1995), “Distinctions, Combinations, and Constraints”, *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knoweldge Sharing*, Menlo Park CA, USA: AAAI Press,.

[Spurr et al. 94] Spurr, K., Layzell, P., Jennison, L., and Richards, N., (1994), *Software Assistance for Business Reengineering*, London: Wiley and Sons.

[Tham et al. 94] Tham, D., Fox, M.S., and Gruninger, M., (1994), “A Cost Ontology for Enterprise Modelling”, *Third Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*, West Virginia University.

[Uschold et a.l 97] Uschold, M., King, M., Moralee, S., Zorgios, Y., (1997), “The Enterprise Ontology”, *Knowledge Engineering Review*, to appear. ftp://ftp.aiai.ed.ac.uk/pub/projects/enterprise/ontology/v1-1-md31-pub.ps.gz

[Vernadat 96] Vernadat, F.B., (1996), “Enterprise Integration: On Business Process and Enterprise Activity Modelling”, *Concurrent Engineering: Research and Applications*, Vol. 4, No. 3, pp. 219-228.

[Yu et al. 96] Yu, E., Mylopoulos, J., and Lesperance, Y., (1996), “AI Models for Business Process Reengineering”, *IEEE Expert*, August 1996, pp. 16-23.