ENTERPRISE ENGINEERING: AN INFORMATION SYSTEMS PERSPECTIVE

Mark S. Fox, Michael Gruninger, Yin Zhan

Department of Industrial Engineering, University of Toronto,
4 Taddle Creek Road, Toronto, Ontario M5S 1A4
tel:1-416-978-6823 fax:1-416-978-3543 internet:{msf,gruninger,zhan}@ie.utoronto.ca

Abstract: We present a framework for enterprise engineering that encompasses the formalization of knowledge in an enterprise, its integration into a software tool, and the visualization of the enterprise. The foundation for the system is the use of generic ontologies for enterprise modelling.

INTRODUCTION

Market competition is forcing firms to reconsider how they are organized to compete. As a basis for change, they are exploring a variety of concepts, including Time-based Competition, Quality Function Deployment, Activity-Based Costing, Quality Circles, Continuous Improvement, Process Innovation, and Business Process Re-Engineering. Regrettably, most of the concepts are descriptive, if not ad hoc, and lack a formal model which would enable their consistent application across firms. Consider business process re-engineering [Davenport 93], [Hammer & Champy 93]. It is very much in the "guild" mold of application; management consultants are the "masters" and they impart their knowledge through "apprenticeship" to other consultants. The knowledge of business process re-engineering has yet to be formalized and reduced to engineering practice.

The goal of the Enterprise Engineering Project at the University of Toronto is to:

1. Formalize the knowledge found in Enterprise Engineering perspectives such as Time-based Competition, Quality Function Deployment, Activity-Based Costing, Quality Circles, Continuous Improvement, Process Innovation, and Business Process Re-Engineering. By formalize, we mean the identification, formal representation and computer implementation of the concepts, methods and heuristics which comprise a particular perspective. This not only enables a precise formulation of the intuitions implicit in practice, but it is also a step towards automating the execution of certain tasks involved in enterprise engineering

2. Integrate the knowledge into a software tool that will support the enterprise engineering function by exploring alternative organization models spanning organization structure and behaviour. The Enterprise Engineering system allows for the exploration of a variety of enterprise designs. The process of exploration is one of design, analysis and re-design, where the system not only provides a comparative analysis of enterprise design alternatives, but can also provide guidance to the designer.

3. Provide a means for visualizing the enterprise from many of the perspectives mentioned above. The process of design is performed through the creation, analysis and modification of the enterprise from within each of the perspective visualizations.

In the remainder of the paper we first provide an overview to the Enterprise Engineering system architecture. We then describe each component in detail. Lastly, we describe the status of our implementation.

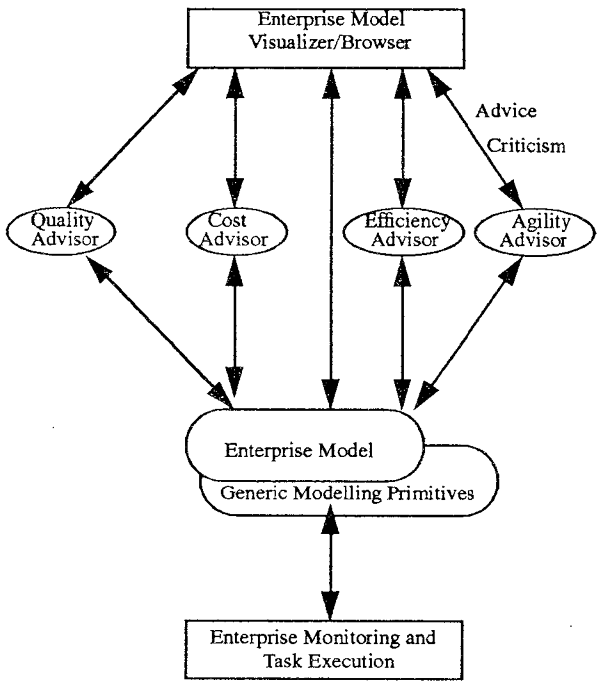
ARCHITECTURE FOR ENTERPRISE ENGINEERING

The Enterprise Engineering system is composed of four main components: its common-sense enterprise model, advisors, visualization, and information agents (see Figure 1).

The foundation for the system is the common-sense enterprise model. It provides a set of generic reusable representations of enterprise knowledge. This includes representations for processes, activities, time, causality, resources, quality, and cost. The enterprise model is used by all other components of the system by providing a shared terminology and set of constraints.

INSTITUTE OF INDUSTRIAL ENGINEERS
3rd Industrial Engineering Research Conference Proceedings

Figure 1



Various perspectives exist in an enterprise, such as efficiency, quality, and cost. Any system for enterprise engineering must be capable of representing and managing these different perspectives in a well-defined way. These ideas are formalized in the notion of advisors that are able to analyze, guide, and make decisions about the current enterprise and possible alternatives.

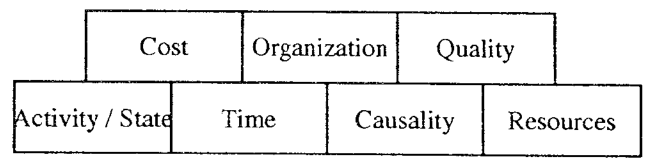
In addition to representing the knowledge in an enterprise, it is useful to be able to visualize the different perspectives that we have in the enterprise. This requires the existence of an environment which can graphically represent the advisors in the system and the interactions among these advisors, including workflow monitoring and execution.

Lastly, there is an execution environment where the portions of the enterprise design may be, i.e., those portions that define databases and machine executable activities, down-loaded for execution by the run-time system.

COMMON SENSE ENTERPRISE MODELLING

Enterprise modelling is an essential component in defining an enterprise.The goal of our enterprise modelling research is to create a generic, reusable representations of Enterprise Knowledge that can be reused across a variety of enterprises. Towards this end, we have been developing the TOVE enterprise ontology [Fox et al 93]. 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. TOVE provides a rich and precise representation of generic knowledge, such as, activities, processes, resources, time, and causality, and of more enterprise oriented knowledge such as cost, quality and organization structure.

Figure 2



Base Ontology

The basic entities in our model are represented as objects with specific properties and relations. Objects are structured into taxonomies. Definitions of objects, attributes and relations are specified in first-order logic, where possible. We then define an ontology in the following way. We first identify the objects in our domain of discourse; these will be represented by constants and variables in our language. We then identify the properties of these objects and the relations that exist over these objects; these will be represented by predicates in our language.

We next define a set of axioms in first-order logic to represent the constraints over the objects and predicates in the ontology. This set of axioms constitutes a microtheory ([Lenat & Guha 90]) and provides a declarative specification for the various tasks we wish to model.

Intuitively, the axioms in the microtheory enable the model to deduce answers to questions that one would normally assume can be answered if one has a "common-sense" understanding of the enterprise. To formalize this intuition we also need to prove results about the properties of our microtheories in order to provide a characterization and justification for our approach; this enables us to understand the scope and limitations of the approach. We use a set of problems, which we call competency questions, that serve to characterize the various ontologies and microtheories in our enterprise model. The microtheories must contain a necessary and sufficient set of axioms to represent and solve these questions. It is in this sense that we can claim to have an adequate microtheory appropriate for a given task, and it is this rigour that is lacking in previous approaches to enterprise engineering.

The following ontologies have been designed as part of the Enterprise Engineering Project at the University of Toronto:

Activities and States: The formalization of the notion of process and activity is crucial in any attempt at business process re-engineering. Activities are the basic events that specify a transformation on the world [Fox et al. 93]. States specify what must be true for an activity to be performed, and what is true once the activity is completed.

Time: Activities are initiated at points in time, and once initiated, they have duration over some interval of time. Further, properties of states hold over the duration of these activities. To represent these intuitions, we extend the temporal relations in [Allen 83].

Resources: All activities require that some objects be available at the time that the activity is performed; this is the motivation for a theory of resources. The various properties that are axiomatized in the microtheory include the distinction between reusable vs. consumable resources, resource commitment, and the availability of resources [Fadel & Fox 93].

Quality: The work in this domain is concerned with creating a terminology that spans quality concepts found in ISO9000, Baldridge Award, etc. [Kim & Fox 93].

Cost: The goal is to formalize the concepts found in activity-based costing [Tham et al 94].

ADVISORS

The best enterprise design is one that optimises each of the perspectives that exist in the enterprise. Examples of enterprise perspectives include: Quality, Cost, Efficiency, Agility, and Incentives. We are developing for each perspective a theory of design that results in the optimization of the perspective. The theory incorporates the ability to measure a partial/complete design and to guide the designer in the decision making. One issue is whether there exists sufficient knowledge of the process of designing and optimizing business activities/processes to incorporate in knowledge-based tools.

To formalize the intuition of design perspectives, we introduce the notion of advisors. An advisor is an encapsulation of one or more micro-theories. It has the ability to analyse, guide and make design decisions. We are currently constructing advisors for Efficiency, Activity-based Costing, Resource Management, and ISO9000; we anticipate advisors for goals and objectives, organization structure, incentives, and culture. In each case, we are defining the tasks, purpose, and responsibilities of the advisor, and represent these tasks using the appropriate ontology. As with the competency questions in the preceding section, each advisor is rigorously characterized by these tasks. This includes specifying what an advisor is analyzing and in what way that they guide (propose different alternatives).

Consider the efficiency advisor. The modelling task provides ontologies that can be used to construct a model of the activities of a process, temporal relations over these activities, and constraints on the usage of resources by the activities. Based on these models the efficiency advisor provides tools to design, analyze and evaluate the enterprise from the perspective of optimising efficiency. For example, it can perform a critical path analysis of an activity graph or process, or it may simulate a process if more complex activity behaviours are involved.

The advisor must be able to represent and model the current status of the process and assess potential changes. This is essential if the advisor is required to guide the designer by presenting alternatives. For example, we may need to know if a process would be more efficient given one ordering of activities rather than another. This may entail identifying the resources that prevent activities from being performed concurrently and thus anticipate resource conflicts that lead to bottlenecks.

Now consider the notion of a ISO9000 advisor, which uses a microtheory of ISO 9003 compliance [Kim & Fox 93]. This microtheory introduces axioms to represent the ISO 9003 requirements and axioms defining how an organization can be ISO 9003 compliant. The primary decision-making capability of the quality advisor is therefore determining whether an organization is ISO 9003 compliant.

The ISO 9003 requirements can be divided into those that can be met by just one process (locally compliant requirements) and those that require several or all processes of an enterprise (globally compliant requirements). For example, to satisfy ISO 9003 local compliance, there must exist processes that perform product identification, inspection and testing, identify test status, control nonconformity, and arrange for handling of products; this is an axiom in the microtheory. An advisor can use this axiom in several different ways. It can be used to analyze a process within the enterprise and decide compliance by verifying the existence of the necessary processes. It can also be used by a designer by recommending the appropriate quality control processes that must be included in order to satisfy local compliance.

ENTERPRISE VISUALIZATION

The "state-of-the-art" of enterprise engineering visualizations is to fill the four walls of a room with pieces of paper detailing the organization structure and the activities within organizational units. Coloured string is used to tie together activities that comprise a process that spans two or more organizational units. The reason why the "four wall" method is preferred is simply the sheer amount of information that has to be "seen" at once and the ease with which it can be modified. Existing design tools, such as Digital's DECModel and Kralman's MAGIC, can only provide small samples of processes and organization structures, hence limiting the user's ability to understand the entirety of the process. In order for computers to play a broader role in enterprise engineering, more creative methods of visualizing enterprises are required. In this section we raise several issues centred around the problem of designing the appropriate visual environment for enterprise engineering, and the role that microtheories and advisors play in any such environment.

Supporting the different levels of description within an enterprise requires appropriate abstraction and elaboration constructs in the underlying knowledge-based system. Aspects of the visualization can also be derived from the ontologies and microtheories of the advisors. The competency questions for these microtheories characterize the necessary properties that they must be able to represent, and it is these properties that are essential for any understanding of the task. Hence these are the important properties that must be displayed in any visualization of the enterprise.

To present the visualization, we may organize the domain knowledge of each individual advisor into a workbench, and provide different viewing projections of a given domain, as well as allow different levels of visualization of a given projection. One objective is the development of a symbology that depicts terms and concepts of the underlying knowledge-based system in a graphical context. The symbology should be precise and general enough to support visual programing for performing the modelling task. These issues are relevant for the efficiency and cost advisors, where the major issue is finding the best visual representation for their tasks and microtheories. They are of particular importance for the more abstract advisors such as goals, objectives, incentives, and culture.

Workflow monitoring and execution are also closely related to the visualization of the enterprise. The problem in this case is to create an adequate representation for the enterprise as it evolves in time, which is captured in the notion of the ontologies for activities, resources, and time. For example, one output of enterprise design is the definition of a graph of activities in which subgraphs of activities define critical enterprise processes. Minimally, we should be able to use the graph to support the monitoring of process performance. Maximally, some of these activities may be executable directly by the system.

This may give the impression that the advisors are independent of each other; this is not the case. Each of the microtheories implemented in the advisors is closely related to the other microtheories, and this leads to a close interdependency among the different perspectives. The status of activities is dependent on the times at which its resources are available; in turn, the cost is dependent on the status of an activity, such as when it is executing and when it has been suspended. All of these relationships provide different perspectives on the processes in an enterprise, as captured in the advisors; however, existing graphical interfaces fail to capture the dimensionality of these interdependencies.

The relationship among the various advisors also leads to the possibility of merging visualizations. How do we represent the interaction between cost and the temporal structure of activities in a process? What role do resource usage constraints play in the quality of a product, and how is this best visualized? Ideally, we should be able to combine the visualizations of each advisor in a way that highlights these interactions.

In the previous section we saw how one of the tasks of the advisors is guiding the designer by representing alternative futures and possibilities for the processes in an enterprise. Although this is done formally by the microtheories of the advisors, there is a need for an adequate visual environment for guiding this kind of hypothetical reasoning.

An additional requirement that arises from this architecture is that an Enterprise Engineering environment must support workgroups - different people will be working on or with the same enterprise model and the system has to coordinate these activities. This also includes the ability to support concurrent modelling processes and to coordinate and monitor the contributions and activities of different people that are involved in these tasks. In turn, this requires the negotiation of inconsistent changes to the model.

ENTERPRISE IMPLEMENTATION

Enterprise execution focuses on the implementation of an enterprise design. In particular, it is concerned with both monitoring the performance of enterprise as specified by the model, and executing tasks that can be automated. Our execution model is based on the Enterprise Management Network [Roboam & Fox 92], and is called the Enterprise Information Architecture (EIA). The execution environment is composed of a set of functional and information agents, each of which performs continuously and autonomously. Functional agents perform activities, and information agents manage the distribution and consistency of information.

Given an enterprise design, we should be able to deduce what the organization structure of the enterprise information system should be, the functional agents and their assigned activities, and how information is to be distributed and maintained across the network in order to support the functional agents. Thus as more and more work becomes automated, the linkage between design and execution increases in importance.

CURRENT STATUS

We have developed ontologies for: activities and states, time, resources, quality, and cost. Advisors and their corresponding microtheories are under development for resource management, activity-based costing and ISO9000 quality compliance. A "virtual factory", called TOVE, has been defined using the ontology, and serves as a testbed for research into enterprise integration. It is implemented in C++ using the ROCK knowledge representation tool from Carnegie Group. The axioms in the various microtheories are implemented using Quintus Prolog which is integrated with the knowledge base in ROCK.

We have also developed a distributed simulation environment called TOVESim which oversees the execution of events and maintains time across multiple agents spread across the internet. We are currently working on extending the ontologies, advisors and the ideas for enterprise visualization discussed in this paper.

RELATED WORK

The foundation for the system is the common-sense enterprise model. It provides a set of generic reusable representations of enterprise knowledge. This includes representations for processes, activities, time, causality, resources, quality, and cost. Other representations have been developed for enterprise knowledge - ICAM [Martin 83], IWI [Scheer 89], PERA [Williams 91]; however, the meanings of the relations used in these representations are ambiguous, and there is no way to know whether this is the appropriate way of representing this knowledge.

We use a set of problems, which we call competency questions, that serve to characterize the various ontologies and microtheories in our enterprise model. The microtheories must contain a necessary and sufficient set of first-order axioms to represent and solve these questions, thus providing a declarative semantics for the system. It is in this sense that we can claim to have an adequate microtheory appropriate for a given task, and it is this rigour that is lacking in previous approaches to enterprise engineering.

[Mi & Scacchi 93] present an environment for software process engineering, including a representation of processes and resources. However, they provide only an operational semantics for their system and the representation of time is not explicit. Although they do provide a query mechanism for properties of objects and relations in their model, it is difficult to characterize the competency of their system.

CONCLUSIONS

In this paper, we have defined an architecture for an enterprise engineering system that allows the exploration of a variety of enterprise designs. This work has raised several important issues. First, it has proposed that the knowledge implicit in engineering practice must be formally represented and characterized through ontologies and microtheories. This formalization provides the foundation for the other components of the system.

In order to integrate this knowledge into a software tool that will support enterprise engineering functions, we introduced the notion of an advisor as a formalization of the different perspectives that we have with respect to an enterprise.

The notion of advisors leads to the architecture presented in Figure 1. The visualization of an enterprise requires a graphical interface that represents the abilities and perspectives of the various advisors as formalized in the advisors' microtheories.

Finally, an important application of the representation of knowledge in ontologies and microtheories is that it enables us to automate the execution of certain enterprise engineering tasks. This also allows us to represent workflow monitoring and execution within the same framework.

ACKNOWLEDGEMENTS

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

REFERENCES

[Allen 83] Allen, J.F. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM.* 26:832-843, 1983.

[Davenport 93] Davenport, T.H. *Process Innovation: Reengineering Work through Information Technology.* Harvard Business School Press, 1993.

[Fadel et al 94] Fadel, F. and Fox, M.S., Gruninger M. A Resource Ontology for Enterprise Modelling (to appear in *Proceedings of the Industrial Engineering Research Conference 1994*).

[Fox et al. 93] Fox, M.S., Chionglo, J., Fadel, F. A Common-Sense Model of the Enterprise, *Proceedings of the Industrial Engineering Research Conference 1993.*

[Hammer & Champy 93] Hammer, M. and Champy J. *Reengineering the Corporation.* Harper Business, 1993.

[Kim & Fox 93] Kim, H. and Fox, M.S. Quality Systems Modelling: A Prospective for Enterprise Integration, *Fourth Annual Meeting of the Production and Operations Management Society.* 1993.

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

[Martin 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.

[Mi & Scacchi 93] Mi, P. and Scacchi, W. *An Environment for Software Process Engineering: Collected Papers.* Information and Operations Management Department, University of Southern California.

[Roboam & Fox 92] Roboam, M. and Fox, M.S. Enterprise Management Network Architecture, *Artificial Intelligence Applications in Manufacturing.* AAAI Press / MIT Press, 1992.

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

[Tham et al 94] Tham D., Fox M.S., Gruninger M. A Cost Ontology for Enteprise Modelling (to appear in *Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises 1994*).

[Williams 91] Williams, T.J., and the Members, Industry-Purdue University Consortium for CIM. *The PURDUE Enterprise Reference Architecture.* Technical Report Number 154, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN 47907, 1991.