Methodology for the Design and Evaluation of Ontologies

Michael Grüninger and Mark S. Fox
Department of Industrial Engineering
University of Toronto
Toronto, Canada M5S 1A4
{ gruninger, msf }@ie.utoronto.ca

April 13, 1995

1 Introduction

As information systems play a more active role in the management and operations of an enterprise, the demands on these systems have also increased. Departing from their traditional role as simple repositories of data, information systems must now provide more sophisticated support to manual and automated decision making; they must not only answer queries with what is explicitly represented in their Enterprise Model, but must be able to answer queries with what is implied by the model. The goal of the TOVE (TOronto Virtual Enterprise) Enterprise Modelling project is to create the next generation Enterprise Model, a Common Sense Enterprise Model. By common sense we mean that an Enterprise Model has the ability to deduce answers to queries that require relatively shallow knowledge of the domain.

We are taking what can be viewed as a 'second generation knowledge engineering' approach to constructing our Common Sense Enterprise Model. Rather than extracting rules from experts, we are 'engineering ontologies.' An ontology is a formal description of entities and their properties, relationships, constraints, behaviours. Through interaction with our industrial partners, we encounter problems that arise in their particular enterprises. Our approach to engineering ontologies begins with using these problems to define an ontology's requirements in the form of questions that an ontology must be able to answer. We call this the competency of the ontology. The second step is to define the terminology of the ontology - its objects, attributes, and relations. In this way the ontology provides the language that will be used to express the definitions in the terminology and the constraints required by the application. The third step is to specify the definitions and constraints on the terminology, where possible. The specifications are represented in First Order Logic and implemented in Prolog. Lastly, we test the competency of the ontology by proving completeness theorems with respect to the competency questions.

Our initial efforts have focused on ontologies to support reasoning in industrial environments. The tasks that we have targeted to support are in 'supply chain management' which extends MRP (Manufacturing Requirements Planning) to include logistics/distribution [Fox and Grüninger 94] and 'Concurrent Engineering' which looks at issues in coordination of engineering design.

Within the Enterprise Engineering project, we are conducting research leading to the creation of an information system to support Enterprise Design (also known as business process reengineering) and Execution. An enterprise design environment allows for the exploration of different designs or models of an enterprise along various perspectives such as efficiency, cost, quality and agility. The axioms formalizing the knowledge in these different design perspectives must be supported by different ontologies.

Much of our effort has been in creating representations of organisation behaviour: activity, state, causality and time [Grüninger and Pinto 95], and the objects they manipulate: resources [Fadel et al. 94], inventory, orders and products. We also have efforts underway in formalising knowledge of ISO 9000 quality [Kim and Fox 95], activity-based costing [Tham et al. 94], organisation [Fox et al. 95], and agility.

For any given ontology, the goal is to agree upon a shared terminology and set of constraints on the objects in the ontology. We must agree on the purpose and ultimate use of our ontologies. We must therefore provide a mechanism guiding the design of ontologies, as well as providing a framework for evaluating the adequacy of these ontologies. Such a framework allows a more precise evaluation of different proposals for an ontology, by demonstrating the competency of each proposal with respect to the set of questions that arise from the applications. These justify the existence and properties of the objects with the ontology. This paper describes the methodology used in the Enterprise Integration Laboratory for the design and evaluation of integrated ontologies, including the proposal of new ontologies and the extension of existing ontologies (see Figure 1). We illustrate these ideas with examples from our activity and organisation ontologies.

2 Motivating Scenarios

The development of ontologies is motivated by scenarios that arise in the applications. In particular, such scenarios may be presented by industrial partners as problems which they encounter in their enterprises. The motivating scenarios often have the form of story problems or examples which are not adequately addressed by existing ontologies. A motivating scenario also provides a set of intuitively possible solutions to the scenario problems. These solutions provide a first idea of the informal intended semantics for the objects and relations that will later be included in the ontology.

Any proposal for a new ontology or extension to an ontology must describe the motivating scenario, and the set of intended solutions to the problems presented in the scenario. This is essential to provide rationale for the objects in an ontology, particularly in cases when there are different objects in different proposals for the same ontology. By providing a scenario, we can understand the motivation for the proposed ontology in terms of its applications.

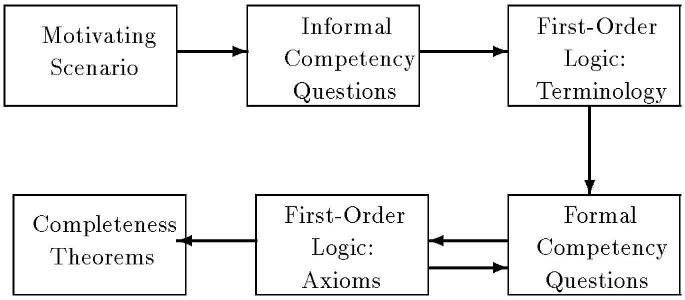

Figure 1: Procedure for Ontology Design and Evaluation

3 Informal Competency Questions

Given the motivating scenario, a set of queries will arise which place demands on an underlying ontology. We can consider these queries to be requirements that are in the form of questions that an ontology must be able to answer. These are the informal competency questions, since they are not yet expressed in the formal language of the ontology.

By specifying the relationship between the informal competency questions and the motivating scenario, we give an informal justification for the new or extended ontology in terms of these questions. This also provides an initial evaluation of the new or extended ontology; the evaluation must determine whether the proposed extension is required or whether the competency questions can already be solved by existing ontologies.

It may happen that people have prior informal ontologies for some application. In this case, for every object, attribute, relation, and axiom in the proposed ontology or proposed extension to an ontology, there must first be an informal competency question, such as a query, which intuitively requires the objects or constraints defined with the object.

Ideally, the competency questions should be defined in a stratified manner, with higher level questions requiring the solution of lower level questions. It is not a well-designed ontology if all competency questions have the form of simple lookup queries; there should be questions that use the solutions to such simple queries.

These competency questions do not generate ontological commitments; rather, they are used to evaluate the ontological commitments that have been made. They evaluate the expressiveness of the ontology that is required to represent the competency questions and to characterize their solutions.

3.1 Activity Ontology

In enterprise modelling, we want to define the actions performed within an enterprise, and define constraints for plans and schedules which are constructed to satisfy the goals of the enterprise. This leads to the following set of informal competency questions:

- Temporal projection – Given a set of actions that occur at different points in the future, what are the properties of resources and activities at arbitrary points in time?

- Planning and scheduling – what sequence of activities must be completed to achieve some goal? At what times must these activities be initiated and terminated?

- Execution monitoring and external events – What are the effects of the occurrence of external and unexpected events (such as machine breakdown or the unavailability of resources) on a plan or schedule?

- Time-based competition – we want to design an enterprise that minimizes the cycle time for a product. This is essentially the task of finding a minimum duration plan that minimizes action occurrences and maximizes concurrency of activities.

3.2 Organisation Ontology

In linking the structure of an organisation with the behaviour of agents within the organisation, we must define how the organisation ontology is integrated with the activity ontology. If we consider organisation to be a set of constraints on the activities performed by agents, then the competency questions for the organisation ontology are extensions of the temporal projection and plan existence problems to incorporate the abilities and obligations of agents. The temporal projection problem is used to characterize the constraints that agents must satisfy to be able to perform activities, and plan existence characterizes the set of achievable goals. We can then propose the following set of informal competency questions for the organisation ontology with respect to agent behaviour, authority, empowerment and commitment, and goal achievement.

- What activities must a particular agent/position/role perform?

- Is it possible for an agent to perform an activity in some situation? That is, does the agent have the ability to perform the activity?

- In order to perform a particular activity, whose permission is needed?

- Is an agent allowed to perform an activity in some situation?

- What goals is an agent committed to achieving?

- What authority constraints are necessary among a set of agents in order to achieve a goal?

- What goals are solitarily unachievable for a given agent? That is, what goals are unachievable using a plan that contains only activities that the agent is capable of performing? Such goals require the assistance of other agents to achieve them.

- What goals are achievable by an agent given the effects of activities that other agents are capable of performing?

- If a goal is solitarily unachievable for a given agent, what agents are required to assist the agent in achieving the goal?

4 Specification in First-Order Logic – Terminology

Once informal competency questions have been posed for the proposed new or extended ontology, the terminology of the ontology must then be specified using first-order logic (or equivalently, in KIF).

Recall that an ontology is a formal description of objects, properties of objects, and relations among objects. This provides the language that will be used to express the definitions and constraints in the axioms. This language must provide the necessary terminology to restate the informal competency questions. If we are designing a new ontology, then for every informal competency question, there must be objects, attributes, or relations in the proposed ontology or proposed extension to an ontology, which are intuitively required to answer the question.

The first step in specifying the terminology of the ontology is to identify the objects in the domain of discourse. These will be represented by constants and variables in the language. Attributes of objects are then defined by unary predicates; relations among objects are defined using n-ary predicates.

4.1 Activity Ontology: Terminology

Within the TOVE project, we have adopted the situation calculus to provide a semantics to our ontology of activity and state. The intuition behind the situation calculus is that there is an initial situation, and that the world changes from one situation to another when actions are per formed. There is a predicate $Poss(a, s)$ that is true whenever an action a can be performed in a situation s.

The structure of situations is that of a tree; two different sequences of actions lead to different situations. Thus, each branch that starts in the initial situation can be understood as a hypothetical future. The tree structure of the situation calculus shows all possible ways in which the events in the world can unfold. Therefore, any arbitrary sequence of actions identifies a branch in the tree of situations.

Further, we impose a structure over situations that is isomorphic to the natural numbers by introducing the notion of successor situation [Reiter 91]. The function $do(a, s)$ is the name of situation that results from performing action s in situation s. We also define an initial situation denoted by the constant S_0.

To represent complex actions, we use the predicate $Do(a, s, s')$ which denotes that if action a is done in situation s, then s' is one of the possible situations reached. A complex action is defined by specifying its subactions and constraints over the occurrence of these subactions.

To define the evaluation of the truth value of a sentence in a situation, we will use the predicate $holds(f, s)$ to represent the fact that some ground literal f is true in situation s. A fluent is a predicate or function whose value may change between situations.

5 Formal Competency Questions

Once the competency questions have been posed informally and the terminology of the ontology has been defined, the competency questions are defined formally as an entailment or consistency problem with respect to the axioms in the ontology. Thus, they will have one of the following forms, where $T_{ontology}$ is the set of axioms in the proposed ontology, T_{ground} is a set of ground literals (instances), and Q is a first-order sentence using only predicates in the language of $T_{ontology}$.

- Determine $T_{ontology} \cup T_{ground} \models Q$
- Determine whether $T_{ontology} \cup T_{ground} \not\models \neg Q$; that is, determine if Q is consistent with $T_{ontology} \cup T_{ground}$.

At this stage, we may not yet have any axioms in $T_{ontology}$; however, the formal competency questions place restrictions on which axioms will be included. It is also important to understand that all terms in the statement of the formal competency questions must be included in the terminology of the ontology.

Every proposal for a new or extended ontology *must* be accompanied by a set of formal competency questions. It is only in this way that we can evaluate the ontology and claim that it is adequate.

Ontologies may be distinguished by the competency questions which they are capable of solving; that is, one ontology may be able to represent and solve a different set of competency questions than another ontology. In this case, the relationship between the ontologies can be formally represented by the questions.

5.1 Problems for Reasoning about Action

Using the terminology presented earlier in the paper, we can define the following formal competency questions for any ontology of activity. In all of the following, T_{action} is the set of axioms in the activity ontology.

Temporal projection is formally defined by the following problem:

Problem 1 *Given a ground formula $\Sigma_{Do}(A, S_0, S_1)$ defining the set of action occurrences, determine*

$$T_{action} \cup \Sigma_{Do}(A, S_0, S_1) \models Q(S_1)$$

for some ground simple formula $Q(S_1)$.

The following problem formalizes the competency question – Is there a sentence characterizing the fluents in a state that guarantee that a plan exists to achieve some goal?

Problem 2 *Let $Q(s)$ be a sentence with no free variables except s. Does there exist a sentence $\Phi(S_0)$ with no free variables such that*

$$T_{action} \models \Phi(S_0) \equiv (\exists s)\, S_0 \leq s \wedge Q(s)$$

6 Specification in First-Order Logic – Axioms

The axioms in the ontology specify the definitions of terms in the ontology and constraints on their interpretation; they are defined as first-order sentences using the predicates of the ontology. It is important to understand the significance of using axioms to define the terms and constraints for objects in the ontology. Simply proposing a set of objects alone, or proposing a set of ground terms in first-order logic, does not constitute an ontology. Axioms must be provided to define the semantics, or meaning, of these terms.

It is also important to realize that this is not the implementation of the ontology; it is the specification of the ontology. However, the implementation of the ontology should be translatable into KIF.

The process of defining axioms is perhaps the most difficult aspect of defining ontologies. However, this process is guided by the formal competency questions. As with the informal competency questions, the axioms in the ontology must be necessary and sufficient to express the competency questions and to characterize their solutions; without the axioms we cannot express the question or its solution, and with the axioms we can express the question and its solutions. Further, any solution to a competency question must be entailed by or consistent with the axioms in the ontology alone. If the proposed axioms are insufficient to represent the formal competency questions and characterize the solutions to the questions, then additional objects or axioms must be added to the ontology until it is sufficient. This development of axioms for the ontology with respect to the competency questions is therefore an iterative process.

There may be many different ways to axiomatize an ontology, but the formal competency questions are not generating these axioms. Rather, we use them to evaluate the completeness of the sets of axioms in any particular axiomatization. In this sense, we can compare the expressiveness of different sets of axioms using the competency questions. If there is a competency question that one set of axioms can represent and another cannot, then the first set is more expressive. If two different axiomatizations can represent a competency question and characterize its solutions, then they are equivalent with respect to the question, and any comparison must use other criteria.

In some applications, there may be a common core ontology that is shared, while different groups use extensions specific to their applications. If this is the case, it is necessary to explicitly characterize the relationships between the core and the different extensions. In fact, the advantage of specifying ontologies in first-order logic is that we are able to represent and reason about the ontological commitments for different applications.

6.1 Theories of Action

The axiomatization is based on the discrete situation calculus [Reiter 91]. The situation calculus is a sorted second order language with equality. There are several domain sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{T}, \mathcal{D}$ for action types, situations, fluents, time, and arbitrary domain objects.

One important property that must be represented is the notion of causality, that is, the specification of what holds in the world after performing some action. As part of the logical specification of the activity ontology, use the solution to the frame problem in [Reiter 91]. The basic idea behind this solution is to derive successor state axioms for each fluent, which provide necessary and sufficient conditions for a fluent to be true in situation $do(a, s)$ given the state in situation s. The successor state axioms have the form

$$(\forall a, s)\ Poss(a, s) \supset [holds(R, do(a, s)) \equiv$$

$$\gamma_R^+(a, s) \vee holds(R, s) \wedge \neg\gamma_R^-(a, s)]$$

where $\gamma_R^+(a, s)$ and $\gamma_R^-(a, s)$ are simple formulae which are used to provide conditions under which an action a produces an effect on a fluent R.

Another important notion is to represent the occurrence of actions. The work of [Pinto and Reiter 93] extends the situation calculus by selecting one branch of the situation tree to describe the evolution of the world as it actually unfolds. This is done using the predicate *actual* defined by the following axioms:

$$actual(S_0) \tag{1}$$

$$(\forall a, s) actual(do(a, s)) \supset actual(s) \wedge Poss(a, s) \tag{2}$$

$$(\forall a_1, a_2, s) actual(do(a_1, s)) \wedge actual(do(a_2, s)) \supset a_1 = a_2 \tag{3}$$

To represent occurrences, we then introduce the predicate $occurs(a, s)$ defined as actions performed along the actual line:

$$(\forall a, s)\ occurs(a, s) \equiv actual(do(a, s)) \tag{4}$$

The notions of the actual line and action occurrences illustrates an important concept with the expressiveness of ontologies. We need to express the following class of constraints: suppose that a plan exists that violates some constraint, but we do not want to allow plans that violate the constraint. How can we distinguish between this constraint and those that must always be satisfied in order for a plan to exist? Using the notion of actual line, we can reason about hypothetical branches where we allow such constraints to be violated, but enforce these constraints on the actual line, so that branches that violate the constraints cannot be actual.

In addition to axioms, it may also be necessary to define sentences that serve as assumptions. These can be used to define special cases of a competency question for which we can provide a solution. For example, in the work on minimum duration, we need the following definitions and assumptions in order to prove a restricted completeness theorem:

Definition 1 *The Action Occurrence Closure (AOC) assumption is the sentence*

$$(\forall a, s)\ occurs(a, s) \equiv a = A_1 \wedge s = S_1 \vee ... \vee a = A_n \wedge s = S_n$$

Definition 2 *The Fluent Duration Closure (FDC) assumption is the sentence*

$$(\forall t, t')\ occurs_T(initiate(F), t) \wedge occurs_T(terminate(F), s'), t') \wedge d = t' - t \supset d = D_1 \vee ... \vee d = D_n$$

Definition 3 *The No Resource Interaction (NRI) assumption is the sentence*

$$(\forall a, a')\ \neg(\exists r)\ shared(r, a, a')$$

7 Completeness Theorems

Once the competency questions have been formally stated, we must define the conditions under which the solutions to the questions are complete. This forms the basis for completeness theorems for the ontology. These theorems have one of the following forms, where $T_{ontology}$ is the set of axioms in the ontology, T_{ground} is a set of ground literals (instances), Q is a first-order sentence specifying the query in the competency question, and Φ is a set of first-order sentences defining the set of conditions under which the solutions to the problem are complete:

- $T_{ontology} \cup T_{ground} \models \Phi$ if and only if $T_{ontology} \cup T_{ground} \models Q$.
- $T_{ontology} \cup T_{ground} \models \Phi$ if and only if $T_{ontology} \cup T_{ground} \cup Q$ is consistent.
- $T_{ontology} \cup T_{ground} \cup \Phi \models Q$ or $T_{ontology} \cup T_{ground} \cup \Phi \models \neg Q$
- All models of $T_{ontology} \cup T_{ground}$ agree on the extension of some predicate P.

Completeness theorems can also provide a means of determining the extendability of an ontology, by making explicit the role that each axiom plays in proving the theorem. Any extension to the ontology must be able to preserve the completeness theorems.

7.1 Theorems for Reasoning about Action

In this section, we present two examples of completeness theorems for the activity ontology. We include them to give some idea of the format for the theorems; for details, we refer readers to the cited papers.

The first example is the completeness theorem for temporal projection from [Grüninger and Pinto 95]. $\mathcal{O}(A, s, s')$ is a set of ordering constraints over subactions of A, and $Circ$ refers to the circumscription of the theory. The second example is a completeness theorem for minimum duration for the special case in which there are no resource constraints. MDA is a set of sentences that formalize the notion of critical path in a set of activities.

Theorem 1 *Any two models $\mathcal{M}, \mathcal{M}'$ of*

$$Circ(T_{action} \cup \Sigma_{Do}(A, s, s') \cup \mathcal{O}(A, s, s'); Do; start)$$

agree on the extension of the predicate holds.

Theorem 2 *Let Σ_{occ} be a set of ground occurs literals. Suppose $\Sigma_{occ} \cup T_{action} \models AOC \wedge FDC \wedge ESC \wedge NRI$ and let $\mathcal{M}$ be a model of $\Sigma_{occ} \cup T_{action}$. Then $\mathcal{M}$ is a minimum duration model of $\Sigma_{occ} \cup T_{action}$ iff $\mathcal{M}$ is a model of $\Sigma_{occ} \cup T_{action} \cup MDA$.*

8 Conclusions

In this paper we have described a methodology for guiding the design of ontologies, as well as providing a framework for evaluating the adequacy of these ontologies. Such a framework allows a more precise evaluation of different proposals for an ontology, by demonstrating the competency of each proposal with respect to the set of competency questions that arise from the applications. These justify the existence and properties of the objects with the ontology. We are then able to prove completeness theorems for the ontologies with respect to the formal competency questions.

References

[1] Fadel, F., Fox, M.S., and Grüninger, M. (1994) A resource ontology for enterprise modelling. *Third Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*, (West Virginia University 1994).

[2] Fox, M.S., Barbuceanu, M., Grüninger, M. (1995) An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour, *Fourth Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*,(West Virginia University 1995).

[3] Fox, M.S. and Grüninger, M. (1994). Ontologies for enterprise integration, *Proceedings of the Second International Conference on Cooperative Information Systems*, pages 82-89.

[4] Grüninger, M. and Pinto, J. (1995) A theory of complex actions for enterprise modelling, *AAAI Spring Symposium: Extending Theories of Action*, Stanford, March 1995.

[5] 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 1995).

[6] Pinto, J. and Reiter, R. (1993) Temporal reasoning in logic programming: A case for the situation calculus. *Proceedings of the Tenth International Conference on Logic Programming.*

[7] Reiter, R. (1991). The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 418-440. Academic Press, San Diego.

[8] Tham, D., Fox, M.S., and Grüninger, M. (1994), A cost ontology for enterprise modelling. *Third Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises*, (West Virginia University 1994).