Designing and Evaluating Generic Ontologies

Michael Grüninger
Department of Industrial Engineering
University of Toronto
gruninger@ie.utoronto.ca

August 28, 2007

1 Introduction

One of the many uses of ontologies has been the modelling of problems and domains in such areas as business process reengineering (where we need an integrated model of the enterprise and its processes, its organisations, its goals, and its customers), in distributed multiagent architectures (where different agents need to communicate and solve problems), and in concurrent engineering and design. We use ontologies for modelling problems and domains. Ontologies are intended to provide an "easy to re-use" library of class objects for modelling the problems and domains. The ultimate goal of this approach is the construction of a library of ontologies which can be reused and adapted to different general classes of problems and environments. However, many ontologies are informal in the sense that they provide have no semantics or axiomatizations. We need a framework for specifying the semantics of object classes and relations in the ontology.

The methodology in this paper supports the formalization of informal ontologies and the integration and reuse of these ontologies. In these cases we have terminology with no or weak axiomatization; the goal is to write sufficient axioms to adequately capture the intended semantics. The fundamental challenge is to give some guidance as to the kinds of axioms an ontological engineer needs to write.

In [Grüninger and Fox 95], we presented a methodology for the design and evaluation of ontologies. 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 within the ontology. In this paper, we refine this methodology to integrate the axiomatizations of the underlying logical theories of an ontology with the notion of object libraries within the ontology. We apply this methodology to the problem of integrating ontologies and to a formal characterization of reusability.

2 Architecture

An ontology is specified by a set of axioms in some formal language. Further, this is not an amorphous set; some axioms are conservative definitions, some are are axiom schemae or metatheoretic constructs expressible using KIF, and some are specifications of object classes and relations. We present an architecture which attempts to make this structure explicit within the axioms in the specification of an ontology. This architecture also integrates the object libraries of an ontology and the theories used to provide the semantics for the terminology in the object libraries.

This architecture is also intended to assist in the axiomatization of an ontology's underlying intuitions of an informal ontology. This includes problems of axiomatizing other informal ontologies when faced with the task of integrating and reusing these ontologies.

At the top level, we have the object classes and relations of the ontology structured as a set of object libraries. All of these classes and relations have definitions expressed using the axiomatization of some set of underlying foundational logical theories. Intermediate between these two levels is a characterization of the classes of sentences which are expressible in the foundational theories and which are used in the definitions of the ontologies' terminology. These classes form the building blocks of the ontologies in the top level of the architecture.

We are consider the situation calculus to be an example of a foundational theory. The building blocks and generic ontologies in the figure are examples of the TOVE ontologies that are defined using the situation calculus. In general, of course, additional foundational theories will be required to provide semantics for ontologies. For example, theories of shape of depiction would be used in an ontology for object recognition, and theories of components would be used for electromechanical product ontologies.

Implicit in the architecture is the fact that there must be some formal language which is used to express the axioms in the foundational theories. This may be first-order logic, second-order logic, modal logic, or possibly some form of nonmonotonic logic.

2.1 Foundational Theories

A foundational theory is a set of distinguished predicates and functions together with some axiomatization. Distinguished predicates are those for which there are no definitions; the intended interpretations of these predicates is defined using the axioms in the foundational theories. We also often use sorted languages to express the theories. These sorts, predicates, and axioms are the ontological commitments of the foundational theory.

All terminology in a generic ontology is defined using classes of sentences in the foundational theories; these classes are the ontology building blocks. Any terminology that does not have a definition is axiomatized in some foundational theory.

2.1.1 Axioms and Model Theory

The semantics for many ontologies are in people's heads; we need some framework for making it explicit. Any ideas that are implicit are a possible source of ambiguity and confusion. The foundational theories provide the semantics for the terminology of an ontology.

Once we have specified a set of axioms in a foundational theory, they can be given an interpretation. Different interpretations can be given, but one of these will be the intended interpretation that guided the development of the axioms. The axiomatization allows a characterization of these interpretations. We can reason about the semantics of the terminology of the ontologies using the models of the axioms in the foundational theories. For example, the situation calculus is used to provide both the set of axioms and the model theory for a suite of ontologies that support reasoning about actions and change.

A model-theoretic account provides a formalization of the notion of motivating scenarios – if a scenario describes a possible state of affairs, then it must be consistent with the axioms of the ontology. The models of an axiomatization provide a way of representing possible states of affairs in a systematic way.

A model-theoretic approach also assists in the development of axioms for terminology. One approach is to define intended interpretations using the predicates in the ontology. By determining which of these interpretations are consistent, we are imposing requirements on any axioms defining these predicates. Constructs in an ontology can be evaluated using counterexamples – to distinguish among different axiomatizations, find falsifying interpretations for each. In this way, we can use model-theoretic arguments to express motivating scenarios, which in turn can be used to intuitively justify axioms and classes of constraints in the ontology.

2.1.2 Comparing Axiomatizations

To compare alternate ontologies requires some common semantic framework [Fillion et al. 95]. One approach is to evaluate different constructs in the ontologies by making their differences explicit within this framework. We need to express the problem of comparing ontologies as one of comparing axiomatizations of the ontologies. The foundational theories of an ontology play this role.

For example, the situation calculus is distinct from the activity ontology. However, it is a formalism that is expressive enough to evaluate, compare, and reason about different proposals for the activity ontology. By interpreting activity ontology terminology within the situation calculus, we have a semantic framework expressive enough to reason about the distinctions among different proposals. An advantage of this approach is that we can use situation calculus and competency questions to compare ontologies through syntactic and semantic restrictions.

This also allows us to compare alternative axiomatizations of a particular problem. We do not force any commitments in an axiomatization; rather, we make the possible commitments explicit.

For example, conditional actions are distinct from actual line constraints defining the triggering of an action by fluents in a state. In the former case we have

$$Do(A, s, s') \equiv (holds(F_1, s) \supset Do(A_1, s, s') \wedge (holds(F_2, s) \supset Do(A_2, s, s')$$

while in the latter case we have

$$holds(F_1, s) \supset occurs(A_1, s)$$

$$(holds(F_2, s) \supset occurs(A_2, s)$$

This constraint can be violated on non-actual branches of the situation tree. In addition, on the actual line, if the fluent F_1 holds in any situation, then action A_1 will occur; intuitively, the action A_1 is triggered by the fluent. In the case of the conditional action, however, A_1 will occur if only if we are performing the action A and F_1 holds in that situation.

Some constructs in different ontologies can be considered as syntactic restrictions on the different sets of constraints in the framework. This allows us to compare alternative ontologies within the language.

2.2 Ontology Building Blocks

Once we have specified the axioms of the foundational theories and characterized these axioms, we can define classes of theories using the predicates and functions in the foundational theories. We call these classes of theories *ontology building blocks*. We explicitly identify building blocks to assist axiomatization; they will be used to provide definitions for the classes and relations of the generic ontologies. All sentences in the generic ontologies belong to the classes of theories defined in the ontology building blocks.

2.2.1 Defining Classes of Theories

Ontology building blocks are specified by classes of theories expressible in the foundational theories. This specification can be either within the language or expressed metatheoretically.

Classes of complex actions can be defined within the language of the situation calculus [Grüninger and Pinto 95]. For example, conditional complex actions are defined by sentences of the form

$$Do(\textbf{if } f \textbf{ then } A_1 \textbf{ else } A_2, s, s') \equiv$$

$$(holds(F, s) \supset Do(A_1, s, s')) \wedge (\neg holds(F, s) \supset Do(A_2, s, s'))$$

Nondeterministic choice complex actions are defined by sentences of the form

$$Do(A_1 \textbf{ or } A_2, s, s') \equiv Do(A_1, s, s') \vee Do(A_2, s, s')$$

Some classes of theories can be given a metatheoretic syntactic characterization. For example, within the situation calculus, goals are expressed by existential state sentences and state constraints are expressed by universal state sentences. Forward expectation occurrence theories are defined by sentences of the form

$$(\forall s)\, \mathcal{O}(A_1, s) \wedge Q(s) \supset (\exists s')\, s < s' \wedge \mathcal{O}(A_2, s')$$

where the occurrence formulae $\mathcal{O}(A_1, s)$ and $\mathcal{O}(A_2, s')$ contain only positive *occurs* literals and $Q(s)$ is a simple state formula.

Some classes of theories can be given characterizations with respect to assumptions that are consistent with or entailed by the theories; we will loosely refer to this as a semantic characterization of the class.

A set of definitions in an ontology uses a building block if the formulae appearing in the definition are in the classes of theories associated with the building blocks.

2.2.2 Design Choices

In specifying definitions for the classes and relations of a generic ontology, we choose from the classes of theories in the ontology building blocks. In this way we can consider the classes of theories to be a set of design choices for a particular generic ontology – each class of theory used in a definition is a choice to specify a definition in a particular way. By explicitly identifying the building blocks used in the definitions of the ontology's terminology, we can characterize the design choices of the ontology.

The characterization of ontology building blocks is based on defining the dimensions in the space of design choices. Along each dimension in this space we have available to us a set of possible classes of theories to specify the definition of a class or relation in the ontology. For example, in specifying the organization ontology, the design dimensions include defining the syntactic classes of the formulae in the theories, defining which sentences are internal and external to a group, and defining which sentences are are single and multiagent theories.

In this sense, design choices are ontological commitments made by the generic ontology. However, it must be clear that building blocks are not augmenting the expressiveness of the underlying foundational theories of the building blocks.

An ontology is equivalent to a set of design choices bundled together as objects. For example, *all* organization ontologies are constructed from the same space of design choices.

How do we justify the adequacy of the dimensions in the space of design choices? That is, how do we show that we have a complete set of design choices? One approach is to use problems/scenarios to identify design choices for the classes in a generic ontology. We can use the notion of design choices as a set of questions that characterize different problems and domains. Each design choice corresponds to a class of theories.

Another approach to ontology design uses the dimensions of the space of theories as questions to guide the writing of definitions for the terminology of the generic ontology. The design choices are represented by different classes of sentences in the building blocks.

2.3 Constructing Ontologies from Building Blocks

We can define objects in an ontology using the building blocks in two ways. In a bottom-up approach, objects are simply different ways of bundling together different classes of theories from the building blocks. For example, within the organization ontology, we identified classes of theories corresponding to the assignment of goals to agents. We use these theories to define classes in the organization ontology such as orders, requests, cooperation, and responses to external events.

In a top-down approach, we define objects in the ontologies and then use the building blocks to define the semantics of these objects and their relations.

In the top-down approach, we may need to extend the building blocks or foundational theories in order to characterize the semantics of terminology. In this case, we determine which classes of sentences are required from the ontology building blocks, and then define how to structure these classes. There are four cases:

1. Sentences in the definitions for the generic ontologies belong to the classes of theories defined in the ontology building blocks.
2. Sentences in the definitions for the generic ontologies do not belong to the syntactic classes defined in the ontology building blocks.
3. Sentences in the definitions for the generic ontologies do not satisfy the assumptions defined in the ontology building blocks.
4. Generic ontologies introduce new sorts into the logical language of a foundational theory.
5. Generic ontologies introduce new distinguished predicates into the logical language of a foundational theory.

In case 1, we are defining new predicates using the classes of formulae in the existing ontology building blocks.

In cases 2 and 3 we are not making any new ontological commitments and we are not extending the expressiveness of the foundational theories; all classes and relations have conservative definitions. In these cases we are finding structure in the space of theories by

- defining new syntactic classes of formulae
- defining new assumptions

In cases 4 and 5 we need to define new axioms for the distinguished predicates in the foundational theories. These new axioms need to be justified with respect to competency questions and characterization theorems for the foundational theories.

2.4 Ontology Libraries

Since the generic ontologies correspond to sets of building blocks, the relationships among the different generic ontologies can defined using the classes of theories in the ontologies' building blocks. This also allows us to characterize the aggregation of ontologies into larger ones and the specialization of ontologies from more generic ontologies, creating a structure over the set of ontologies with respect to genericity. Aggregation of ontologies is defined by set inclusion of building blocks. The genericity ordering among ontologies can be defined the logical relationships among the building blocks. The classes of theories in the building the blocks of a more generic ontology will be stronger than the classes of theories in the building blocks of a less generic ontology. In this way, we can define the expressiveness of an ontology by comparing the classes of theories in its building blocks.

3 Competency Questions for Generic Ontologies

The competency questions in [Grüninger and Fox 95] address the expressiveness of a foundational theory, in particular, the situation calculus. These competency questions are used to characterize distinguished predicates; however, this only applies to the foundational theories. In this section, we use the competency question methodology for designing and evaluating generic ontologies. In particular, we show how the characterization theorems for generic ontologies are related to the characterization theorems for the foundational theories.

The motivating scenarios and competency questions are used to characterize the classes of theories required to define terms in the building blocks for the generic ontologies. In this way the competency questions for generic ontologies can be used to define the design choices.

To characterize generic ontologies, we prove theorems about the classes of theories used as the building blocks for the ontologies. The theorems characterizing the axioms characterize solutions to reasoning problems as special cases defined using the classes of theories in the building blocks. Since generic ontologies are constructed using classes of sentences in the foundational theories, they inherit the solution to the competency questions from the foundational theories. For example, this is related to the problem of characterizing plan existence for process plans and partial schedules, i.e. reasoning problems for specific classes of theories.

This characterization also provides a template or a questionnaire with which we can specify classes of objects in an ontology. For example, the proof that an object library for the activity ontology is complete must follow from the characterization theorems for the theory of complex actions [Grüninger and Pinto 95]. We prove that the object library is complete with respect to those properties that all complex actions must satisfy.

The classes of theories used to define ontology building blocks are analogous to classes of differential equations. With differential equations, we first characterize the classes of equations (axioms in the ontology) and then we characterize the solutions to these equations (declarative specification of solutions to the reasoning problems for some class of theories).

To justify the set of differential equations for a particular problem, we find the class of equations that matches the problem characteristics. Analogously, to justify the set of design choices in an ontology ,we find the classes of theories that match the characteristics of the reasoning problems in the competency questions.

4 Integrating Ontologies

We can use the architecture of Figure **??** to support the integration of ontologies. It is important to realize that there are two distinct forms of integration. The first occurs when combining ontologies that have been designed for the same domain, such as when we are integrating legacy systems. The other form of integration occurs when we are combining ontologies from different domains, such as process and product ontologies.

The primary problem with integration is that two ontologies may use the same terminology but with different semantics. Using the notion of ontology building blocks we can resolve this problem by considering the classes of theories used to provide definitions for the ontologies rather than the terminology of the ontologies themselves. The difficulty of integration corresponds to the degree to which the ontologies share their building blocks and foundational theories. There are three cases:

Integration of building blocks In this case, the ontologies share the same set of building blocks, but use them to define different terminology. Integration can be achieved by combining the definitions, introducing new terms in cases of overloading.

Integration of foundational theories In this case, the ontologies do not share all of their building blocks. However, since they do share their foundational theories, all ontologies can extend their set of building blocks by defining new classes of theories.

If the sets of building blocks for each ontology are consistent, then we can integrate the ontologies by combining these sets. The building blocks of the new integrated ontology may be structured since the classes of theories for one ontology may be stronger than the classes of theories for other ontologies.

If the sets of building blocks for each ontology are mutually inconsistent, then we cannot integrate the ontologies. The best that we can do in this case is to identify the subset of building blocks that can be shared by all ontologies, and use this as a basis for partial integration.

Ontology translation If the ontologies do not share foundational theories, integration will be difficult. A better option in this case would be translation among the ontologies using the notion of relative interpretation; this will be discussed in the next section.

5 Characterizing Reusability

To be effective, ontologies must support reusability, so that we can import and export models among different software systems. The problem is that when software tools are applied to new domains, they may not perform as expected, since they relied on assumptions that were satisfied in the original applications but not in the new ones. By characterizing classes of domains and tasks within these domains, ontologies provide a framework for determining which aspects of an ontology are reusable between different domains and tasks.

In this section, we provide a formal characterization of reusability among different problems through the notion of reducibility of competency questions.

5.1 Defining Domains

A domain can be characterized by the classes of objects and constraints that are required to specify competency questions and their solutions. Within the TOVE framework, this corresponds to identifying the design decisions used to represent the questions. Syntactically, these design decisions are represented as the sets of ontology building blocks.

Reusability of an ontology with respect to domains is therefore characterized by identifying the building blocks of the ontology with the design decisions that define the domain.

5.2 Specification of Competency Questions

Suppose that we are given a set of competency questions which are not specified using the terminology in the existing ontology. In order to determine whether the existing ontology can represent these questions, we must provide some form of translation for the new terminology. In general, we can consider this new terminology to be part of another ontology, and the problem is equivalent to the translation from the existing ontology to the target ontology.

There are three ways of demonstrating the reusability of an ontology with respect to translations among sets of different competency questions. In each of these cases, we must demonstrate that the competency questions from the target ontology can be represented using the native ontology?

Reuse of ontology building blocks This approach requires axioms specifying the definition of terms in the target ontology using the existing ontology building blocks.

Recall that formal competency questions 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 $T_{ontology} \cup T_{ground} \not\models \neg Q$

In the current case, both ontologies share the same language, so that we can represent both sets of competency questions with respect to the same sentence Q.

Let $T_{ontology}$ be the set of axioms in the existing ontology, and let $T'_{ontology}$ be the set of axioms in the target ontology. In order to provide the translation, we will require a new set of axioms, T_{def}, which provides definitions for the predicates and functions in $T'_{ontology} \cup T'_{ground}$ using the Ontology Building Blocks from the native ontology.

We can define the reduction of competency questions as the following metatheoretic problem:

$$T'_{ontology} \cup T'_{ground} \models Q \Rightarrow T_{ontology} \cup T_{ground} \cup T_{def} \models Q$$

Reuse of foundational theories In this approach, we identify the target ontology's building blocks using the existing ontology's foundational theories. This requires axioms specifying new sets of ontology building blocks corresponding to the target ontology.

The approach for reducing competency questions in this case is identical to the previous case. The only additional difficulty is that the axioms in T_{def} require the use of new Ontology Building Blocks from the target ontology.

Ontology translation If the foundational theories of the ontologies cannot be reused, then we must provide a relative interpretation of the foundational theories of the two ontologies. This approach requires the specification of an interpretation of the existing foundational theory into the target foundational theory. This is the most difficult case, since the sentences used to express the different sets of competency questions use different languages.

We must first define a relative interpretation π of T into T' ([Enderton 72]). We must then prove that the interpretation π is faithful.

Suppose π is a faithful interpretation of the theory $T_{ontology} \cup T_{def}$ into the theory $T'_{ontology}$. We can define the reduction of competency questions as the following metatheoretic problem:

$$T'_{ontology} \cup T'_{ground} \models Q \Rightarrow T_{ontology} \cup T_{ground} \cup T_{def} \models \pi[Q]$$

5.3 Characterization Theorems

Proving the competency of an ontology with respect to a given competency question may require the introduction of assumptions that must be satisfied by the axioms of the ontology. Different solutions to the competency question can then be characterized by these sets of assumptions.

Recall that characterization 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.

The sentence Φ in these statements of characterization theorems plays the role of the assumptions. Reusability of an ontology with respect to these solutions is characterized by specifying the relationship between these assumptions and constraints satisfied by problems in a given domain. In particular, this requires the identification of any assumptions that are "built-in" the axioms of the ontology, since such assumptions restrict the reusability of the ontology to domains which do not satisfy the same assumptions.

6 Conclusions

In this paper we have described an architecture for integrating the axiomatizations of the underlying logical theories of an ontology with the notion of object libraries within the ontology. This approach allows us to specify the semantics of object classes and relations in the ontology. It also provides formal foundations to the problems of integrating ontologies and characterizing the reusability of ontologies among different domains.

References

[1] Enderton, H. (1972) *A Mathematical Introduction to Logic.* Academic Press.

[2] Fillion, F., Menzel, C., Blinn, T., and Mayer, R. (1995) An ontology-based environment for enterprise model integration, *Workshop on Basic Ontological Issues in Knowledge Sharing*, International Joint Conference on Artificial Intelligence 1995, Montreal.

[3] 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).

[4] Grüninger, M. and Fox, M.S. (1995) Methodology for the design and evaluation of ontologies, *Workshop on Basic Ontological Issues in Knowledge Sharing*, International Joint Conference on Artificial Intelligence 1995, Montreal.

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