Ontologies and Domain Theories

Michael Grüninger
Department of Mechanical and Industrial Engineering
University of Toronto
gruninger@mie.utoronto.ca

Abstract

Although there is consensus that a formal ontology consists of a set of axioms within some logical language, there is little consensus on how a formal ontology differs from an arbitrary theory. There is an intuitive distinction between the formal ontology and the set of domain theories that use the ontology, but there has been no characterization of this distinction. In this paper we utilize the notions of definable sets and types from model theory mathematical logic to provide a semantic characterization of the domain theories for an ontology. We illustrate this approach with respect to several formal ontologies from mathematical logic and knowledge representation.

Motivation

Ontological engineering was born with the promise of reusability, integration, and interoperability. Of increasing importance are the problems merging ontologies from different domains and translating among multiple ontologies from the same domain. However, to a large degree, we have not yet delivered these promised benefits. What we lack is a framework within which people can develop and share reusable ontologies.

On the one hand, formal ontologies are specific theories – we are not defining new languages or logics. On the other hand, formal ontologies are different from arbitrary theories in that we intuitively think of ontologies as being the reusable portion of domain theories. Of course, this begs the question of defining domain theories, and it raises the perennial debate of the difference between ontologies and knowledge bases.

In the course of providing a formal characterization of domain theories for ontologies, we are guided by several intuitions.

- Domain theories and queries are constructed using ontologies – typical reasoning problems include sentences that describe a particular scenario in addition to the axioms of the ontologies.
- Ontologies are the reusable parts of domain theories, in the sense that all domain theories for an ontology are extensions of a unique set of axioms in the ontology.
- In semantic interoperability scenarios, software applications exchange sentences that are written using ontologies, rather than exchange axioms from the ontologies themselves.

The objective of this paper to is to provide a semantic characterization of domain theories, that is, one that is based on properties of the models of the formal ontology.

Some Motivating Examples

We consider several ontologies and the sentences that are intuitively their domain theories. We begin with two mathematical theories which are well understood before moving on to two ontologies from the knowledge representation community.

Algebraically Closed Fields Suppose that two software applications share the ontology of algebraically closed fields (Hodges 1993), for example, CAD software that is based on algebraic geometry. Such software applications will exchange shapes that are specified by polynomials; they are not exchanging axioms in the ontology. In this case, we can see that the domain theories for algebraically closed fields are polynomials.

Groups Domain theories for the ontology of groups (Hodges 1993) are either explicitly specifying particular groups or subgroups of other groups. A group presentation defines a group by specifying a set of elements of a group (known as generators) such that all other elements of the group can be expressed as the product of the generators subject to a set of equations (known as relations among the generators). For example, the group presentation for the cyclic group of order three is the equation $a \cdot a \cdot a = 1$.

Time Ontologies Consider an ontology of time T^{dense} (Hayes 1996) in which the set of timepoints is linearly ordered and dense Such an ontology is typically used to specify the underlying constraints in commonsense reasoning problems about events (e.g. "Bob left home before arriving at work and Alice arrived at work after Bob"). This set of constraints constitutes a domain theory for the ontology T^{dense}; in general, the domain theories consist of boolean combinations of sets of timepoints that form intervals on the linear ordering.

Situation Calculus The axiomatization of situation calculus in (Reiter 2001) includes a set of foundational axioms (the ontology) together with a set of axioms which plays the role of a domain theory.

A simple state formula is a formula which contains a unique situation variable and which contains only *holds* literals. A precondition axiom for an activity A is a sentence of the form

$$(\forall s)\, poss(A, s) \supset Q(s)$$

where $Q(s)$ is a simple state formula. An effect axiom for an activity A is a sentence of the form

$$(\forall s)\, Q_1(s) \supset holds(F, do(A, s))$$

where $Q(s)$ is a simple state formula and F is a fluent. Basic action theories, which consist of sets of precondition and effect axioms, are domain theories for situation calculus.

Domain Theories and Definable Sets

The characterization of ontologies and domain theories rests on the model-theoretic notion of definability. After introducing this notion, we will use it to distinguish between the different classes of theories within an ontology.

Definable Sets

We will adopt the following definition from (Marker 2002):

Definition 1 *Let $\mathcal{M}$ be a structure in a language L.*

A set $X \subseteq M^n$ is definable in $\mathcal{M}$ iff there is a formula $\varphi(v_1, ..., v_n, w_1, ..., w_m)$ of L and $\overline{\mathbf{b}} \in M^m$ such that

$$X = \{\overline{\mathbf{a}} \in M^n \;:\; \mathcal{M} \models \varphi(\overline{a}, \overline{b})\}$$

X is A-definable if there is a formula $\psi(\overline{v}, w_1, ..., w_l)$ and $\overline{\mathbf{b}} \in A^l$ such that

$$X = \{\overline{\mathbf{a}} \in M^n \;:\; \mathcal{M} \models \varphi(\overline{\mathbf{a}}, \overline{\mathbf{b}})\}$$

We say that X is $\emptyset$-definable if $A = \emptyset$. If A is nonempty, we say that X is definable with parameters.

Example 1 *Suppose $\mathcal{M}$ is an algebraically closed field.*

The set of even numbers is $\emptyset$-definable in $\mathcal{M}$:

$$\{x \;:\; (\exists y)\, x = y + y\}$$

The set of prime numbers is $\emptyset$-definable in $\mathcal{M}$:

$$\{x \;:\; (\forall y, z)\, (y \cdot z = x) \supset (y = x) \vee (z = x)\}$$

The set

$$\{x \;:\; a_0 + a_1 x + a_2 x^2 + ... + a_n x^n = 0\}$$

is definable with parameters $a_0, ..., a_n$.

Definitional Extensions and Core Theories

An ontology is specified by a set of axioms in some formal language. Nevertheless, this is not an amorphous set, and the notion of definability allows us to distinguish between different kinds of sentences within an ontology.

Definition 2 *A theory T_1 is a definitional extension of a theory T iff every constant, function, and relation in models of T_1 is $\emptyset$-definable in models of T.*

It is easy to see that a definitional extension of a theory T is also a conservative extension of T, although the converse is not true; that is, there are conservative extensions of theories which are not definitional extensions.

Definition 3 *A theory T^{core} is a core theory iff it is not a definitional extension of any other theory.*

Combining these two classes of sentences gives us the following definition of an ontology.

Definition 4 *An ontology T^{onto} is a theory consisting of a set of core theories and a set of definitional extensions.*

Intuitively, the core theories axiomatize the primitive functions and relations in the ontology. If a core theory in an ontology is an extension of some other core theories in the ontology, then it is a nonconservative extension.

In the case of the PSL Ontology ((Gruninger 2004), (Bock & Gruninger 2005), (Gruninger & Kopena 2004)), the conservative extensions within the ontology are axiomatizations of invariants that are used to classify the models of the core theories within the ontology.

Domain Theories

We are still faced with the question of how domain theories are different from the other two classes of theories within an ontology. Whereas a definitional extension is an axiomatization of the $\emptyset$-definable sets in a model of an ontology T^{onto}, we will say that a domain theory for an ontology T^{onto} is an axiomatization of sets that are definable with parameters in some model of T^{onto}.

Definition 5 *A theory T^{dt} is a domain theory for an ontology T^{onto} iff every sentence in T^{dt} defines a set $X \subseteq M^n$ with parameters in some model $\mathcal{M}$ of T^{onto}.*

In general, domain theories are not conservative extensions of the ontology. For example, the domain theory consisting of the equation

$$(\mathbf{a} \cdot (\mathbf{a} \cdot \mathbf{a})) = 1$$

in the theory of groups entails the sentence

$$(\exists x, y)\, (x \cdot y) = (y \cdot x)$$

which is not entailed by the axioms in the theory of groups alone.

On the other hand, domain theories are distinct from arbitrary nonconservative extensions of the ontology. For example, the sentence

$$(\forall x, y)\ (x \cdot y) = (y \cdot x)$$

axiomatizes abelian groups; it forms a nonconservative extension of the theory of groups, yet we would not consider it to be a domain theory.

Domain Theories and Types

The next step is to show how the set of domain theories for an ontology can be characterized with respect to properties of the models of the ontology. This will allow us to formalize the intuitions presented earlier in the motivation.

Types

Types describe a model of a theory from the point of view of a single element or a finite set of elements ((Marker 2002), (Rothmaler 2000)).

Definition 6 *Let $\mathcal{M}$ be a model for a language $\mathcal{L}$.*

The type of an element $\mathbf{a} \in M$ is defined as $type_{\mathcal{M}}(\mathbf{a}) := \{\phi : \phi$ is a formula of $\mathcal{L}, \mathcal{M} \models \phi\ \}$

An n-type for a theory T is a set $\Phi(x_1, ..., x_n)$ of formulae, such that for some model $\mathcal{M}$ of T, and some n-tuple $\overline{a}$ of elements of $\mathcal{M}$, we have $\mathcal{M} \models \phi(\overline{a})$ for all ϕ in Φ.

If t is an n-type, then a model $\mathcal{M}$ realizes t iff there are $a_1, ..., a_n \in M$ such that

$$\mathcal{M} \models t(a_1, ..., a_n)$$

A type p is a complete n-type iff $\phi \in p$ or $\neg\phi \in p$ for any formula ϕ with n free variables; a partial type is a type that is not complete.

Informally, the type for an element in a model is a set of formulae which are satisfied by some set of elements in the model. An n-type for a theory is a consistent set of formulae (each of which has n free variables) which is satisfied by a model of the theory.

Characterization Theorems for Domain Theories

The model-theoretic notion of type allows us to formalize the intuition that domain theories are theories about elements in the domain of a model of the ontology.

Theorem 1 *A set of sentences T^{dt} is a domain theory for an ontology T^{onto} iff it is logically equivalent to a boolean combination of finite partial n-types for T^{onto}.*

Proof: $\Rightarrow$:) Let $\varphi(x_1, ..., x_n)$ be a sentence in a domain theory for T^{onto} and let

$$\{\overline{a}\ :\ \mathcal{M} \models \varphi(\overline{x})\}$$

be the set defined by this sentence in a model $\mathcal{M}$ of T^{onto}. It is easy to see that this set realizes the finite n-type $\varphi(x_1, ..., x_n)$ in $\mathcal{M}$.

$\Leftarrow$:) The set of elements that realize a finite type in $\mathcal{M}$ constitute a definable set. The boolean combination of finite partial n-types is equivalent to the union, intersection, complement, and projection of definable sets, and these operations preserve definable sets. Therefore, the boolean combination of n-types is logically equivalent to a domain theory. □

This result shows that we can specify all possible domain theories for an ontology by identifying the finite partial types for elements in the models of the ontology.

Not all types correspond to domain theories, since a type that consists of an infinite set of formulae may not be first-order definable. For example,

$$\{0 < c, S(0) < c, S(S(0)) < c, ...\}$$

is an infinite type that is realized by a nonstandard number $\mathbf{c}$ in a model of $Th(\mathbb{N}, 0, S, <)$, yet the set is not first-order definable in the theory.

The next two theorems characterize domain theories with respect to the models of the ontology, and formalize the intuition that ontologies are the reusable parts of domain theories.

Theorem 2 *If T^{dt} is a domain theory for an ontology T^{onto} then there exists a model $\mathcal{M}$ of T^{onto} such that*

$$T^{onto} \cup T^{dt} \subseteq Th(\mathcal{M})$$

Proof: By Definition 5, the sentences in T^{dt} define sets with parameters in some model $\mathcal{M}$ of T^{onto}. We therefore have

$$T^{onto} \subseteq Th(\mathcal{M})$$

Suppose that there is a sentence $\Sigma \in T^{dt}$ such that $\Sigma \not\subset Th(\mathcal{M})$. In this case, we must have $\mathcal{M} \models \neg\Sigma$, which would mean that Σ does not define a set in $\mathcal{M}$, and hence would not be a sentence in a domain theory. We therefore also have

$$T^{dt} \subseteq Th(\mathcal{M})$$

□

From this result we can see that models of of a domain theory are models of the ontology.

Theorem 3 *For any model $\mathcal{M}$ of T^{onto}, there exists a domain theory T^{dt} for T^{onto} such that*

$$T^{onto} \cup T^{dt} \subseteq Th(\mathcal{M})$$

Proof: Since $\mathcal{M}$ is a model of T^{onto}, we have

$$T^{onto} \subseteq Th(\mathcal{M})$$

If T^{dt} is the set of sentences that define sets in $\mathcal{M}$, then $T^{dt} \neq \emptyset$ (since finite sets are definable). T^{dt} is therefore a domain theory such that

$$\mathcal{M} \models T^{dt}$$

As a result, we know that $T^{onto} \cup T^{dt}$ is consistent. Since $\mathcal{M} \models T^{onto} \cup T^{dt}$, we have

$$T^{onto} \cup T^{dt} \subseteq Th(\mathcal{M})$$

□

Note that any definable set must have some axiomatization, whereas nondefinable sets cannot be axiomatized by any theory. Furthermore, every model contains definable sets (since finite sets are always definable). Consequently, domain theories will always exist for any ontology.

We can define a complete domain theory as one that satisfies

$$T^{onto} \cup T^{dt} = Th(\mathcal{M})$$

for some model $\mathcal{M}$ of T_{onto}. In other words, a complete domain theory is an axiomatization of a particular model of the ontology. Not all ontologies will have complete domain theories

Techniques for Specifying Domain Theories

Model theory provides several techniques for specifying the types for first-order theories. The most widely use technique is known as the elimination of quantifiers, in which one focusses on the sets that are definable by formulae that are quantifier-free.

A theory T admits the elimination of quantifiers if for every formula ϕ there is a formula ψ such that

$$T \models \phi \equiv \psi$$

One typically determines this by specifying a set quantifier-free formulae Δ (known as the elimination set) such that for every formula ϕ in the language of T there is a formula ψ which is a boolean combination of formulae in Δ, and ϕ is equivalent to ψ in every model of T. It is easy to see that in ontologies that admit elimination of quantifiers, the elimination set characterizes the set of types.

Revisiting the Examples

The set of types for many ontologies within mathematical logic have been specified for within the literature. We can see that the types for the ontologies that we considered for the theories in the motivation do indeed correspond to the intuitions that we have about their domain theories.

Algebraically Closed Fields and Polynomials Since algebraically closed fields admit the elimination of quantifiers, it can be shown ((Marcja & Toffalori 2003)) that any irreducible polynomial corresponds to a complete 1-type and that 2-types correspond to algebraic curves. In other words, there is a one-to-one correspondence between the set of roots of polynomials (algebraic numbers) and definable elements in the models of T_{field}. There is also the complete 1-type that is realized by all numbers that are transcendental over models of the ontology; this type is not generated by a finite set of formulae.

Presentations and Groups Although the theory of groups does not admit elimination of quantifiers, it can be shown that all 1-types for T_{group} are of the form

$$(\exists y, z)\, x = y \cdot z$$

We can see that both presentations and group equations are domain theories for groups, since they are boolean combinations of 1-types. In a sense, the presentation is equivalent to the types realized by all elements of the group G; when a presentation exists, it is a complete axiomatization of the theory $Th(G)$ for the group.

Time Ontologies Models of T_{dense} are isomorphic to dense linear orderings, whose n-types have been fully characterized in (Rosenstein 1973). The n-types for T_{dense} are therefore boolean combinations of literals of the form $before(v_i, v_j)$ and $v_i = v_j$. Thus the types for dense linear orderings correspond to the domain theories discussed in Section 1.1.

Action Theories in Situation Calculus Although there has been no work on the characterization of the types for $T_{sitcalc}$ we can still show that action theories define sets in models of $T_{sitcalc}$, and so are domain theories for $T_{sitcalc}$.

The precondition axiom for each action $\mathbf{a}$ is realized by the definable set of situations

$$\{\mathbf{s_1} \,:\, \mathbf{s_1} = \mathbf{do}(\mathbf{a}, \mathbf{s}), \langle \mathbf{s_1} \rangle \in \mathbf{executable}\}$$

that is, the set of executable situations that correspond to occurrences of $\mathbf{a}$. The effect axiom for each action $\mathbf{a}$ is realized by the definable set of situations

$$\{\mathbf{s_1} \,:\, \mathbf{s_1} = \mathbf{do}(\mathbf{a}, \mathbf{s}), \langle \mathbf{f}, \mathbf{s_1} \rangle \in \mathbf{holds} \Leftrightarrow \langle \mathbf{f}, \mathbf{s} \rangle \notin \mathbf{holds}\}$$

that is, the set of situations that achieve or falsify specific fluents. A complete characterization of all types and domain theories for $T_{sitcalc}$) is an open research problem.

Evaluating the Ontology

We can evaluate the correctness and completeness of the ontology and domain theories with respect to the characterization of definable sets. For correctness, all domain theories for an ontology must be consistent with the ontology. For completeness, we need to determine if we construct models of the ontology that do not realize any types corresponding to some class of domain theories.

Definition 7 *Let Σ be a set of types for a theory T.*

T is definably complete with respect to Σ iff every model of T realizes some type in Σ.

In $T_{sitcalc}$, precondition axioms are domain theories, but not all activities realize precondition axioms i.e. there are other classes of domain theories

Theorem 4 *The ontology $T_{sitcalc}$ is not definably complete with respect to the set of basic action theories.*

Proof: We will construct a model of $T_{sitcalc}$ that does not satisfy any basic action theory (i.e. set of precondition and effect axioms).

let $\mathbf{s_1}, \mathbf{s_2}$ be situations in the situation tree that agree on state, that is, for any fluent $\mathbf{f}$,

$$\langle \mathbf{f}, \mathbf{s_1} \rangle \in \mathbf{holds} \Leftrightarrow \langle \mathbf{f}, \mathbf{s_2} \rangle \in \mathbf{holds}$$

Now specify the extension of the **poss** relation for an activity **a** such that

$$\langle \mathbf{a}, \mathbf{s_1} \rangle \in \mathbf{poss}, \langle \mathbf{a}, \mathbf{s_2} \rangle \notin \mathbf{poss}$$

The activity **a** cannot realize any precondition axiom, since the same simple state formula is realized by both $\mathbf{s_1}$ and $\mathbf{s_2}$.

Now specify the extension of the **holds** relation for the activity **a** such that

$$\langle \mathbf{f}, \mathbf{do}(\mathbf{a}, \mathbf{s_1}) \rangle \in \mathbf{holds}, \langle \mathbf{f}, \mathbf{do}(\mathbf{a}, \mathbf{s_2}) \rangle \notin \mathbf{holds}$$

The activity **a** cannot realize any effect axiom, since the same simple state formula is realized by both $\mathbf{s_1}$ and $\mathbf{s_2}$. □

On the other hand, the PSL Ontology explicitly axiomatizes the classes of activities that realize the types corresponding to basic action theories[1]

Theorem 5 *Let MAA (Markovian Activity Assumption) be the sentence*

$$(\forall a) activity(a) \supset markov\_precond(a) \wedge markov\_effect(a)$$

The ontology $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore} \cup MAA$ *is definably complete with respect to the set of basic action theories.*

It should be noted that $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$ alone is not definably complete, since there are models that do not realize precondition and effect axioms; on the other hand, all models of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore} \cup MAA$ realize precondition and effect axioms.

We can also use this approach to show that several approaches to ontologies are in fact specifying classes of domain theories rather than ontologies.

This is prevalent in approaches to reasoning about change that simply add a temporal argument to the set of relations.

Furthermore, one cannot specify domain theories using axiom schemata, since there will typically be mutually inconsistent domain theories for the same ontology.

Classifying Domain Theories

We can use the notion of definable completeness of an ontology to classify the domain theories for the ontology. In particular, we can classify domain theories with respect to the sets that are $\emptyset$-definable by the sentence Φ such that $T^{onto} \cup \Phi$ is definably complete with respect to the domain theories.

For example, by Theorem 5, $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore} \cup MAA$ is definably complete; activities in the set defined by the sentence MAA realize the types corresponding to basic action theories. Activities that are not in the set (that is, activities that do not satisfy the sentence MAA) do not realize the types corresponding to basic action theories. This gives a model-theoretic definition of basic action theories, rather than simply a syntactic definition.

Reasoning Problems

Many reasoning problems with ontologies (such as decision problems for mathematical theories) incorporate domain theories as well as the set of axioms in the ontologies themselves.

The Word Problem in group theory is specified for a particular group and it requires both the axioms for groups as well as the presentation for the group:

$$T_{group} \cup \Sigma_{presentation} \models (w = 1)$$

The query in this case is the product of group elements w.

In a temporal reasoning problem, we consider a particular scenario of temporal constraints in addition to the axioms for the time ontology, and determine whether or not a particular temporal constraint is entailed by the scenario:

$$T_{time} \cup \Sigma_{scenario} \models before(T_1, T_2)$$

For situation calculus, the antecedent of a reasoning problem such as planning includes basic action theories, while the query sentence is an existentially quantified simple state formula:

$$T_{sitcalc} \cup \Sigma_{action} \models (\exists s)\, Q(s)$$

In general, an entailment problem for an ontology T^{onto} has the form

$$T^{onto} \cup \Sigma^{dt} \models \Sigma^{query}$$

where Σ^{dt} is a domain theory for T^{onto} and Σ^{query} is a sentence in the language of the ontology. This leads to the next question – what class of sentences in the language of the ontology characterize the query?

Any sentence that is a query (that is, a sentence in Σ_{query}) can also be considered to be a domain theory. For example, in the word problem for groups, the query sentence is a group equation, which is a type for the theory of groups. Similarly, simple state formulae are types for fluents in situation calculus.

We can provide a model-theoretic characterization of queries using the following notion:

Definition 8 *A type p is isolated iff there is a formula* $\varphi \in p$ *such that for any* $\psi \in p$*, we have*

$$T \models (\forall \overline{v})\, \varphi(\overline{v}) \supset \psi(\overline{v})$$

Queries therefore correspond to nonisolated types for the ontology. Using this definition, we can also consider queries to be weak domain theories, in the sense that they are entailed by other domain theories. We can therefore apply the earlier techniques for arbitrary domain theories to provide a characterization of the possible queries in reasoning problems that use a particular ontology.

[1] The axiomatization of *markov_precond* in CLIF (Common Logic Interchange Format) can be found at `http://www.mel.nist.gov/psl/psl-ontology/part42/state_precond.def.html`
The axiomatization of *markov_effect* in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part42/state_effects.def.html`

The same techniques that were used to characterize all possible domain theories for an ontology by specifying the types for the ontology can be used to characterize the queries by specifying the nonisolated types for the ontology. We can also classify the queries for an ontology by characterizing the additional sentences that are required in order for an ontology to be definably complete with respect to the class of queries.

Summary

Although there is an intuitive distinction between the formal ontology and the set of domain theories that use the ontology, there has been no characterization of this distinction. In this paper we have utilized the notions of definable sets and types from model theory mathematical logic to provide a semantic characterization of the domain theories for an ontology that gives a clear logical distinction between ontologies and domain theories.

Domain theories for an ontology are the axiomatization of definable sets in models of the ontology. This is equivalent to saying that a domain theory for an ontology is a boolean combination of finite partial n-types for the ontology.

This characterization of domain theories serves as an evaluation criterion for ontologies, which can in turn be used to classify the domain theories for an ontology.

This lays the groundwork for a comprehensive methodology for the evaluation of formal ontologies by specifying the complete sets of n-types that are realized in models of the ontologies.

References

Bock, C., and Gruninger, M. 2005. PSL: A semantic domain for flow models. *Software and Systems Modeling* 4:209–231.

Gruninger, M., and Kopena, J. 2004. Semantic integration through invariants. *AI Magazine* 26:11–20.

Gruninger, M. 2004. Ontology of the Process Specification Language. In Staab, S., and Studer, R., eds., *Handbook of Ontologies in Information Systems*. Springer-Verlag.

Hayes, P. 1996. A catalog of temporal theories. Technical Report UIUC-BI-AI-96-01, University of Illinois.

Hodges, W. 1993. *Model Theory*. Cambridge University Press.

Marcja, A., and Toffalori, C. 2003. *A Guide to Classical and Modern Model Theory*. Kluwer Academic Publishers.

Marker, D. 2002. *Model Theory: An Introduction*. Springer-Verlag.

Reiter, R. 2001. *Knowlege in action: logical foundations for specifying and implementing dynamical systems*. Cambridge, MA, USA: MIT Press.

Rosenstein, J. 1973. *Linear Orderings*. Academic Press.

Rothmaler, P. 2000. *Introduction to Model Theory*. Taylor and Francis.