Specifying Ontology Design Patterns with an Ontology Repository

Michael Grüninger and Megan Katsumi

Department of Mechanical and Industrial Engineering, University of Toronto
Toronto, Ontario, Canada M5S 3G8
gruninger@mie.utoronto.ca, katsumi@mie.utoronto.ca

Abstract. Within the Common Logic Ontology Repository (COLORE), the notion of reducibility among ontologies has been used to characterize relationships among ontologies. This paper uses techniques such as relative interpretation to show how one set of ontologies within the repository can be reused to characterize the models of other ontologies that are used in a wide variety of domains. A central theme of the paper is that ontology design patterns can be formalized as core ontologies within the ontology repository.

1 Introduction

The COLORE (Common Logic Ontology Repository) project[1] is building an open repository of first-order ontologies that serve as a testbed for ontology evaluation and integration techniques, and that can support the design, evaluation, and application of ontologies in first-order logic. The logical relationships among the set of first-order ontologies in the repository can also be used as basis for the verification of an ontology with respect to its intended models as well as decomposition of ontologies into modules.

We will show how COLORE follows the vision of ontology design patterns as proposed in [6] and [5]. Different notions of ontology design patterns have been used, ranging from syntactic criteria to structural properties of ontologies. As a result, several methodological questions remain challenges – How can we evaluate ontology design patterns and their application? How are design patterns reused? Within COLORE, design patterns are formalized as core ontologies within the repository. Patterns are reused via the metatheoretic relationships of relative interpretation and definable equivalence. In this sense, the ontology design patterns within COLORE are semantic (model-theoretic) rather than syntactic. On the other hand, the approach described in this paper can also be used to generate axioms for new ontologies, in which case we can consider core ontologies to serve as syntactic templates for axioms.

After an informal discussion of ontology design patterns in the context of COLORE, we give an overview of the relationships between ontologies within COLORE. The notions of relative interpretation, definable equivalence, and reduction play a key role in formalizing the reuse of ontologies. In particular, these notions give us techniques for evaluating ontology design patterns and proving that a pattern is correctly and completely exemplified by a set of ontologies. We will illustrate this approach using sets of ontologies within COLORE.

[1] http://code.google.com/p/colore/source/browse/trunk/

2 COLORE and Ontology Design Patterns

In general, Ontology Design Patterns (OPs) are meant to serve as reusable solutions for various aspects of ontology design [6], and the structure of the ontologies in COLORE and the relationships defined between them can provide similar support. COLORE provides a means of sharing content ontology design patterns (CPs) while providing solutions that address specific instances of some of the modelling problems that other OPs are designed to solve.

Of the six families of OPs recognized in [6], the Structural, Correspondence, and Content families of OPs have strong parallels in COLORE:

Structural OPs include what are referred to as Logical and Architectural OPs. Architectural OPs represent possible structures for an ontology being designed. These structures are meant to assist with design choices when computational complexity is a concern, and also to serve as reference material to guide designers in creating their own structures. In particular, *external* Architectural OPs provide patterns for ontology modularization, ("meta-level constructs"). Examples of these *external* Architectural OPs can be found in COLORE as each ontology is stored in modules[9] that are connected to form the ontology using the imports relation.

Correspondence OPs include what are referred to as Reengineering and Mapping OPs. Mapping OPs provide a means to describe the relationship(s) that exist between elements in different ontologies. Similarly, relationships are defined between the terms used in different ontologies in COLORE. In this way the relationships represent specific instances of Mapping OPs. Relationships between ontologies themselves are also described so that users may compare their semantics; these relationships are based on the notion of reducibility discussed in the following section.

Content OPs (CPs) appear to be the most widely used family of OPs. They are typically domain oriented and provide axioms that are intended to be reused as "building blocks" in order to construct an ontology. CPs can also serve other functions in ontology development such as evaluation. Although they are not necessarily domain-oriented, we view the *core theories* of COLORE to be examples of useful CPs, as all ontologies in COLORE are reducible to sets of these ontologies. Using the notion of *intended models*, the core theories in COLORE can also be used for ontology verification ([11],[8]).

3 Relationships between Ontologies in COLORE

The sets of ontologies within COLORE are organized based on the notion of the reduction of one ontology to a set of ontologies. In this section, we review the background for understanding reduction and the role it plays in organizing ontologies within the repository.

3.1 Relative Interpretation

The notion of interpretability between theories[2] is widely used within mathematical logic and applications of ontologies for semantic integration [14]. We will adopt the definition of relative interpretation from [4], in which the mapping π is an interpretation of a theory T_1 with language. L_1 into a theory T_2 with language L_2 iff it preserves the theorems of T_1.

Definition 1. *An interpretation π of a theory T_1 into a theory T_2 is faithful iff*

$$T_1 \not\models \sigma \Rightarrow T_2 \not\models \pi(\sigma)$$

for any sentence $\sigma \in \mathcal{L}(T_1)$.

Thus, the mapping π is a faithful interpretation of T_1 if it preserves satisfiability with respect to T_1. We will also refer to this by saying that T_1 is faithfully interpretable in T_2.

Definable equivalence is a generalization of the notion of logical equivalence to theories that do not have the same signature.

Definition 2. *Two theories T_1 and T_2 are definably equivalent iff T_1 is faithfully interpretable in T_2 and T_2 is faithfully interpretable in T_1.*

For example, the theory of timepoints is definably equivalent to the theory of linear orderings. On the other hand, although the theory of partial orderings is faithfully interpretable in the theory of timepoints, these two theories are not definably equivalent, since the theory of timepoints is not interpretable in the theory of partial orderings.

Definition 3. *Let T_0 be a theory with signature $\Sigma(T_0)$ and let T_1 be a theory with signature $\Sigma(T_1)$ such that $\Sigma(T_0) \cap \Sigma(T_1) = \emptyset$.*
Translation definitions for T_0 into T_1 are sentences in $\Sigma(T_0) \cup \Sigma(T_1)$ of the form

$$\forall \overline{x}\, p_i(\overline{x}) \equiv \Phi(\overline{x})$$

where $p_i(\overline{x})$ is a relation symbol in $\Sigma(T_0)$ and $\Phi(\overline{x})$ is a formula in $\mathcal{L}(T_1)$.

Translation definitions can be considered to be an axiomatization of the interpretation of T_0 into T_1. As noted in the previous section, the use of translation definitions in COLORE is similar to Mapping OPs, insofar as they specify relationships between terms used in different ontologies in order to compare their semantics.

[2] In this paper, we consider an ontology to be a set of first-order sentences (axioms) that characterize a first-order theory, which is the closure of the ontology's axioms under logical entailment.

The non-logical lexicon (signature) of a first-order theory T, denoted by $\lambda(T)$, is the set of all constant symbols, function symbols, and relation symbols that are used in T.

The language of T, denoted by $\mathcal{L}(T)$, is the set of all first-order formulas that only use the non-logical symbols in the signature $\lambda(T)$.

3.2 Hierarchies

If an ontology is characterized by its set of ontological commitments, then such commitments will be formalized by sets of axioms. Moreover, in order for the commitments to be comparable, their axiomatizations need to be expressed in the same language. Using these intuitions, we can define an ordering over a set of theories:

Definition 4. *A hierarchy* $\mathbb{H} = \langle \mathcal{H}, < \rangle$ *is a partially ordered, finite set of ontologies* $\mathcal{H} = T_1, ..., T_n$ *such that*

1. $\mathcal{L}(T_i) = \mathcal{L}(T_j)$*, for all* i, j*;*
2. $T_1 \leq T_2$ *iff*

$$T_1 \models \sigma \Rightarrow T_2 \models \sigma$$

for any $\sigma \in \mathcal{L}(T_1)$*.*

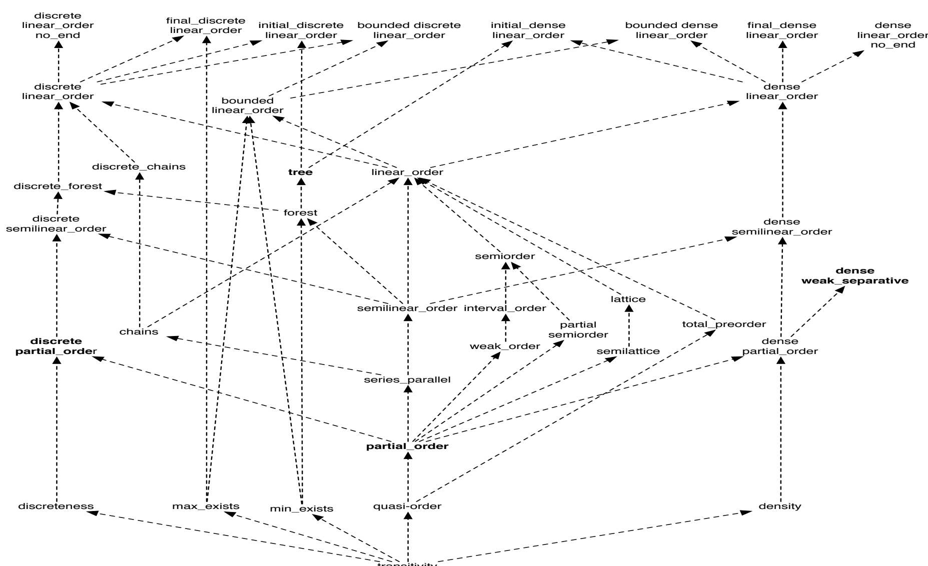

Fig. 1. Ontologies in $\mathbb{H}^{ordering}$: the core hierarchy of orderings. Dashed lines denote nonconservative extension. Theories in bold are ones which are used in this paper.

The theories within two hierarchies in COLORE are shown in Figures 1 and 2. The Ordering Hierarchy[3] contains ontologies that axiomatize different classes of orderings, such as partial orderings, linear orderings, trees, and lattices.

The Mereology Hierarchy[4] contains ontologies that axiomatize different intuitions related to the concept of parthood (see [15] for a full discussion of these ontologies).

[3] http://code.google.com/p/colore/source/browse/trunk/ontologies/core/ordering
[4] http://code.google.com/p/colore/source/browse/trunk/ontologies/core/mereology

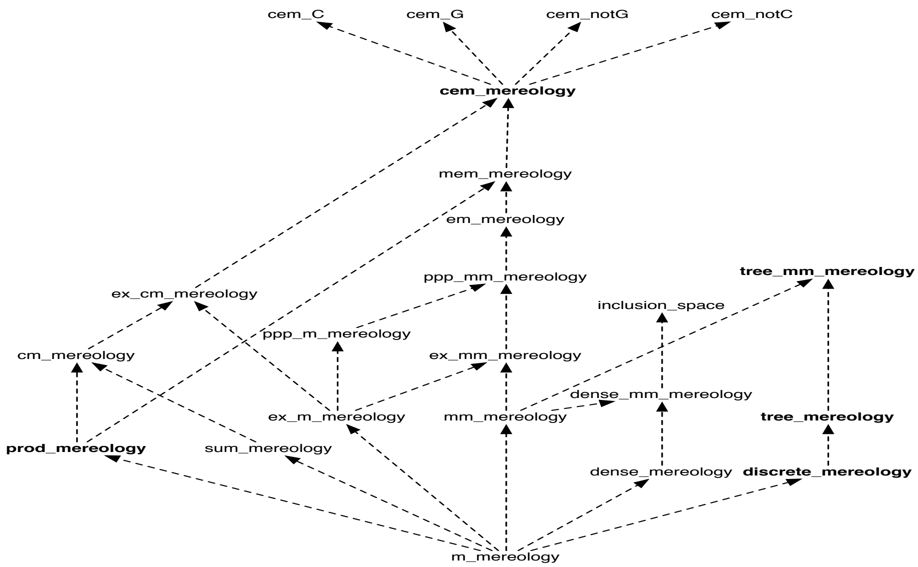

Fig. 2. Ontologies in $\mathbb{H}^{mereology}$: the hierarchy of mereologies. Dashed lines denote nonconservative extension. Theories in bold are ones which are used in this paper.

Note that all extensions of ontologies in the same hierarchy are nonconservative. An ontology T is a root ontology iff it is not the extension of any other ontology in the same hierarchy. Within the $\mathbb{H}^{ordering}$ Hierarchy, the root ontology is the axiomatization of a transitive relation. Within the $\mathbb{H}^{mereology}$ Hierarchy, the root ontology is the axiomatization of a basic mereology (in which the parthood relation is transitive, reflexive, and antisymmetric). This ontology is definably equivalent to the theory $T_{partial\_order}$ within the $\mathbb{H}^{ordering}$ Hierarchy.

3.3 Reducibility

Definable equivalence is a relationship between two ontologies; we can generalize this to a relationship among sets of ontologies. The basis for this approach is the model-theoretic notion of reducibility introduced in [8].

Definition 5. *A ontology T is reducible to a set of ontologies $T_1, ..., T_n$ iff*

1. *T faithfully interprets each T_i, and*
2. *$T_1 \cup ... \cup T_n$ faithfully interprets T.*

We will also refer to the set of ontologies $T_1, ..., T_n$ in the definition as the reduction of T in the repository.

It is easy to see that two definably equivalent ontologies are reducible to each other. For example, within COLORE, the ontology $T_{mereology}$ is reducible to the ontology $T_{linear\_ordering}$ and vice versa.

The following result from [9] characterizes the relationship between reducibility and definable equivalence, and it will be used in this paper to prove results about reducibility:

Theorem 1. *Let $T_1, ..., T_n$ be a set of ontologies such that $\Sigma(T_i) \cap \Sigma(T_j) = \emptyset$ for all $1 \leq i, j \leq n, i \neq j$.*

A ontology T is reducible to $T_1, ..., T_n$ iff T is definably equivalent to $T_1 \cup ... \cup T_n$.

Section 4 will present the reductions of several different ontologies, and discuss their relationship to design patterns.

3.4 Core and Complex Hierarchies

The notion of the reducibility of ontologies can be used to specify an ordering on the set of hierarchies.

Definition 6. *Let $\mathbb{H}_1, ..., \mathbb{H}_n$ be a finite set of hierarchies.*

A repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ is a partially ordered set where

- $\mathcal{R} = \{\mathbb{H}_1, ..., \mathbb{H}_n\}$*;*
- $\mathbb{H}_i \sqsubseteq \mathbb{H}_j$ *iff each root ontology in* $\mathbb{H}_j$ *has a reduction that contains a ontology* T *in* $\mathbb{H}_i$*.*

For example, we can show that $\mathbb{H}_{ordering} \sqsubseteq \mathbb{H}_{mereology}$, since the root ontology in $\mathbb{H}_{mereology}$ is definably equivalent to the ontology $T_{partial\_ordering}$ in $\mathbb{H}_{ordering}$. On the other hand, $\mathbb{H}_{mereology} \not\sqsubseteq \mathbb{H}_{ordering}$, since the root ontology for $\mathbb{H}_{ordering}$ (which is $T_{transitive}$) is not reducible to any ontology in $\mathbb{H}_{mereology}$.

Since we are dealing with repositories that contain a finite set of hierarchies, we are guaranteed that the partial ordering $\sqsubseteq$ has minimal elements.

Definition 7. *A hierarchy $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ is a core hierarchy iff it is a minimal hierarchy in the repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$.*

An ontology T is a core ontology theory iff it is in a core hierarchy.

A complex hierarchy $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is a hierarchy which is not minimal in the repository $\langle \mathcal{R}, \sqsubseteq \rangle$.

An ontology T is a complex ontology iff it is in a complex hierarchy.

Through the notion of reducibility, we can see that core ontologies play the role of building blocks for all other ontologies within the repository. A complex ontology is either constructed from a set of core ontologies or it is an ontology that imposes additional ontological commitments on a core ontology (e.g. the root theory of theory of the $\mathbb{H}^{mereology}$ Hierarchy imposes additional ontological commitments that make the parthood relation reflexive and antisymmetric). If the repository contains multiple equivalent core hierarchies, then the reduction will contain multiple definably equivalent core ontologies, and hence there might exist multiple reductions that contain different sets of core ontologies.

Within COLORE, the notion of a core ontology is therefore based on the logical notion of reducibility, rather than on the distinction between generic vs domain ontologies, as in [13].

4 Hierarchies as Design Patterns

Core ontologies within the repository can be definably equivalent to multiple ontologies in other hierarchies. In this sense, they play the role of design patterns that are reused to verify other ontologies; that is, they can be used to prove that the intended models of an ontology are isomorphic to the models of the axiomatization of the ontology. In this section, we consider in detail one set of core ontologies and show how its relationships to a surprising variety of other ontologies from remarkably different domains.

4.1 Subposet Hierarchy

Each ontology in the Subposet Hierarchy[5] is an extension of an ontology from the Mereology Hierarchy and an ontology from the Ordering Hierarchy.

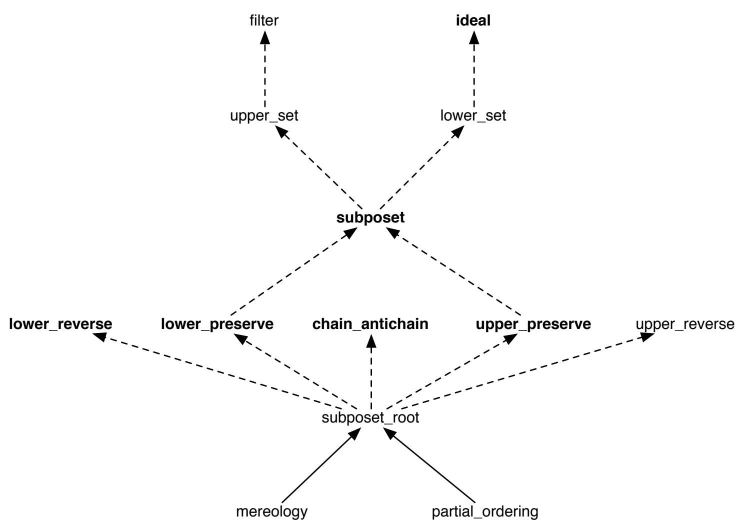

Fig. 3. Ontologies in $\mathbb{H}^{subposet}$: the hierarchy of theories of relationships between partially ordered sets. Dashed lines denote nonconservative extension and solid lines denote conservative extension. Ontologies in bold are ones which are used in this paper.

The ontologies shown in Figure 3 form the basis for the $\mathbb{H}^{subposet}$ Hierarchy. The root ontology $T_{subposet\_root}$ is the union of $T_{m\_mereology}$ and $T_{partial\_ordering}$, and is a conservative extension of each of these ontologies. Thus, each model of $T_{subposet\_root}$ (and hence each model of any ontology in the hierarchy) is the amalgamation of a mereology substructure and a partial ordering substructure.

The ontologies shown in Figure 3 contain additional axioms that constrain how the mereology is related to the partial ordering. In models of $T_{subposet}$, the mereology is a

[5] http://code.google.com/p/colore/source/browse/trunk/ontologies/core/subposet/

subordering of the partial ordering. T_{ideal} strengthens this condition by requiring that the mereology is a subordering of the partial ordering which forms an ideal. In models of $T_{chain\_antichain}$, elements that are ordered by the mereology are not comparable in the partial ordering.

All ontologies within the $\mathbb{H}^{subposet}$ Hierarchy combine one of the ontologies in Figure 3 together with one of the ontologies in Figure 2 and one of the ontologies in Figure 1. In the following sections, we will explore how different ontologies in the $\mathbb{H}^{subposet}$ Hierarchy serve as design patterns.

4.2 Multimereology Hierarchy

Motivated by biomedical ontologies such as GALEN and Foundational Model of Anatomy, Bittner and Donnelly ([2],[3]) have investigated a class of ontologies that combine different kinds of mereological relations. In particular, they axiomatized three relations for part, component, and containment in an ontology which they call T_{fo_pcc}. The subtheory for the $part\_of$ relation has models which are isomorphic to a dense mereology with the weak supplementation principle. Models of the subtheory for the $component\_of$ relation are isomorphic to discrete mereologies which satisfy the weak supplementation principle as well as what Bittner and Donnelly refer to as the no-partial-overlap property – if x and y are distinct overlapping objects, then either x is a part of y or y is a part of x. Finally, models of the subtheory for the $contained\_in$ relation are isomorphic to a discrete partial ordering. Models of T_{fo_pcc} are amalgamations of the models of the three subtheories, and they are referred to as parthood-component-containment structures. Within the COLORE repository, these theories appear in the $\mathbb{H}^{multimereology}$ Hierarchy.

The ontology for parthood-component-containment structures also contains three axioms that specify how the substructures are combined. The component-of structure is a subordering of the parthood structure, while the relationship between containment and parthood satisfies the following two conditions – parts are contained in the container of the whole and that if a part contains something then so does the whole.

Bittner and Donnelly give an informal description of the models of their ontology, but do not provide a complete characterization of the models up to isomorphism. We can, however, use theories within the $\mathbb{H}^{subposet}$ Hierarchy to verify [6] that the models of the ontologies are isomorphic to the intended models of Bittner and Donnelly.

Theorem 2. *T_{fo_pcc} is definably equivalent to*

$$(T_{tree\_mm\_mereology} \cup T_{dense\_weak\_separative} \cup T_{subposet})$$
$$\cup (T_{dense\_mm\_mereology} \cup T_{discrete\_mereology} \cup T_{lower\_preserve} \cup T_{upper\_preserve}).$$

In this sense, we can prove that an ontology design pattern is correctly exemplified for given ontology O by proving that the core ontology is definably equivalent to O.

We can also use definable equivalence to extract multiple design patterns from the same ontologies in those cases where an ontology can be decomposed into modules.

[6] The proofs for all theorems can be found in http://stl.mie.utoronto.ca/colore/subposet-theorems.pdf

Recognizing that T_{fo_pcc} is actually definably equivalent to two different ontologies in the $\mathbb{H}^{subposet}$ Hierarchy, we can specify two ontologies, T_{ppcmp} and T_{ppcnt}, which form a modular decomposition of T_{fo_pcc}.

Theorem 3. *T_{ppcmp} is definably equivalent to the ontology*
$$T_{tree\_mm\_mereology} \cup T_{dense\_weak\_separative} \cup T_{subposet}$$

Theorem 4. *T_{ppcnt} is definably equivalent to*
$$T_{dense\_weak\_separative} \cup T_{discrete\_mereology} \cup T_{lower\_preserve} \cup T_{upper\_preserve}$$

It is important to note that T_{ppcmp} and T_{ppcnt} are each definably equivalent to a unique ontology within the $\mathbb{H}^{subposet}$ Hierarchy. As stated in the previous section, each ontology within the $\mathbb{H}^{subposet}$ Hierarchy is a combination an ontology in the $\mathbb{H}^{ordering}$ Hierarchy, an ontology in $\mathbb{H}^{mereology}$ Hierarchy, and one of the “building block" ontologies in Figure 3 that specifies how the mereology and partial ordering are amalgamated.

4.3 Periods Hierarchy

The axioms in the ontologies of the $\mathbb{H}^{periods}$ Hierarchy[7] were first proposed by van Benthem in [1]. The key ontology of this hierarchy, referred to as T_{period}, constitutes the minimal set of conditions that must be met by any period structure and has two relations (*precedence* and *inclusion*) and two conservative definitions (for the *glb* and *overlaps* relations) as its signature. Transitivity and irreflexivity axioms for the precedence relation make it a strict partial order, and transitivity, reflexivity, and antisymmetry axioms for the inclusion relation make it a partial order; the axioms of monotonicity enforce correct interplay between the precedence and inclusion relations. Van Benthem further includes an axiom that guarantees the existence of greatest lower bounds between overlapping intervals.

Theorem 5. *T_{period} is definably equivalent to the ontology*
$$T_{prod\_mereology} \cup T_{partial\_ordering} \cup (T_{upper\_preserve} \cup T_{lower\_reverse} \cup T_{chain\_antichain}).$$

The relationships between the ontologies in this hierarchy were explored in [12]. In particular, additional theories within the $\mathbb{H}^{subposet}$ Hierarchy were shown to be definably equivalent to various extensions of T_{period} as axiomatized by van Benthem. This illustrates how we can use design patterns to specify the axiomatization of new ontologies in a hierarchy. Conversely, a subtheory of T_{period} was used to identify a new ontology within the $\mathbb{H}^{subposet}$ Hierarchy, thus illustrating how we can abstract new design patterns from a set of existing ontologies.

[7] http://code.google.com/p/colore/source/browse/trunk/ontologies/complex/periods

4.4 Subactivities in the PSL Ontology

The PSL Ontology uses the $subactivity$ relation to capture the basic intuitions for the composition of activities. This relation is a discrete partial ordering, in which primitive activities are the minimal elements.

The core ontology[8] $T_{subactivity}$ alone does not specify any relationship between the occurrence of an activity and occurrences of its subactivities. For example, we can compose $paint$ and $polish$ as subactivities of some other activity, say $surfacing$, and we can compose $make\_body$ and $make\_frame$ into another activity, say $fabricate$. However, this specification of subactivities alone does not allow us to say that $surfacing$ is a nondeterministic activity, or that $fabricate$ is a deterministic activity.

The primary motivation driving the axiomatization of T_{atomic} is to capture intuitions about the occurrence of concurrent activities. Since concurrent activities may have preconditions and effects that are not the conjunction of the preconditions and effects of their activities, concurrency in models of T_{atomic} is represented by the occurrence of one concurrent activity rather than multiple concurrent occurrences.

Atomic activities are either primitive or concurrent (in which case they have proper subactivities). The core ontology[9] T_{atomic} introduces the function $conc$ that maps any two atomic activities to the activity that is their concurrent composition. Essentially, what we call an atomic activity corresponds to some set of primitive activities – every concurrent activity is equivalent to the composition of a set of primitive activities. Although $T_{subactivity}$ can represent arbitrary composition of activities, the composition of atomic activities is restricted to concurrency.

Theorem 6. *The ontology $T_{subactivity} \cup T_{atomic\_act}$ is definably equivalent to the ontology*
$T_{cem\_mereology} \cup T_{discrete\_partial\_ordering} \cup T_{ideal}$

By this Theorem, models of $T_{subactivity} \cup T_{atomic\_act}$ are isomorphic to a structure in which a mereological field (on the set of atomic activities) forms an ideal within a discrete partial ordering (on the set of all activities).

4.5 Occurrence Trees in the PSL Ontology

Within the PSL Ontology, an occurrence tree[10] is a partially ordered set of activity occurrences, such that for a given set of activities, all discrete sequences of their occurrences are branches of the tree. An occurrence tree contains all occurrences of *all* activities; it is not simply the set of occurrences of a particular (possibly complex) activity. Because the tree is discrete, each activity occurrence in the tree has a unique successor occurrence of each activity.

In addition, there are constraints on which activities can possibly occur in some domain. Although occurrence trees characterize all sequences of activity occurrences, not all of these sequences will intuitively be physically possible within the domain.

[8] http://code.google.com/p/colore/source/browse/trunk/ontologies/complex/psl/subactivity
[9] http://code.google.com/p/colore/source/browse/trunk/ontologies/complex/psl/atomic
[10] http://code.google.com/p/colore/source/browse/trunk/ontologies/complex/psl/occtree

We will therefore want to consider the subtree of the occurrence tree that consists only of *possible* sequences of activity occurrences; this subtree is referred to as the legal occurrence tree.

Theorem 7. *The ontology $T_{pslcore} \cup T_{occtree}$ is definably equivalent to the ontology $T_{tree\_mereology} \cup T_{tree} \cup T_{ideal}$*

By this Theorem, models of $T_{pslcore} \cup T_{occtree}$ are isomorphic to a structure in which a tree mereology (on the set of legal activity occurrences) forms an ideal within a tree ordering (on the set of all activity occurrences). It is interesting to notice that T_{ideal} is used both for this ontology as well as for $T_{subactivity}$, demonstrating how one core ontology can be reused as a pattern across very different generic ontologies.

5 Discussion Points

In the previous section we provided examples of the ways in which core ontologies within COLORE can be utilised as CPs. We showed how a variety of real-world ontologies were comprised of core theories from the same hierarchy, and how even the same core theories were reused in different ontologies. We also demonstrated how core ontologies could be used to verify that an ontology contained the desired CPs (core theories), and how new core ontologies (CPs) could be identified by abstracting from ontologies in COLORE.

The ontologies in COLORE's hierarchies (specifically the core theories) correspond well to the definition of CPs provided by [6]: *"CPs are small ontologies that mediate between use cases (problem types) and design solutions. They are used as modelling components: ideally, an ontology results from a composition of CPs, with appropriate dependencies between them, plus the necessary design expansion based on specific needs"*. Based on this definition, each ontology in COLORE could be considered to be a CP as any of the modules could conceivably be reused to build other ontologies. However, in this paper we have focused on the core theories as they are most recognizable as CPs - although they are not necessarily domain-oriented, they are definably equivalent to theories that appear in multiple, different domains. They serve as syntactic templates for axioms in a variety of domains, so in a sense they combine aspects of CPs with the more domain-independent Logical OPs.

In this paper we have also explored the way in which the relationships defined in COLORE may be considered OPs, as the assistance they provide for ontology development is similar to the aid provided by OPs. Nevertheless, some of the features of COLORE offer capabilities beyond what is currently offered by the OP community. For example, because of the formalized nature of the relationships specified in COLORE, automated reasoning can be implemented to verify the mappings between the ontologies. In addition, the notion of reducibility can be implemented to identify useful CPs from ontologies in COLORE – the more theories that are reducible to a particular core theory, the more useful it is. Automated theorem provers may also be used to verify that an ontology is in fact a core theory. Lastly, OPs were not intended to be restricted to a particular representation language [6] and the use of first-order logic in COLORE supports this ideal as patterns from a wide range of languages may be represented.

We should emphasize that we do not believe that COLORE can or should replace traditional OPs. Although there are aspects of OPs for which COLORE offers similar solutions, there are also OPs which are completely absent from the relationships in COLORE. We believe that COLORE offers useful perspectives on OPs that may be beneficial to the OP community. In the other direction, much attention has been paid to promoting the use of OPs (CPs specifically); the results of this may be useful for the future development of COLORE. In particular, we can learn from the use of generalized use cases and competency questions to aid in users in the reuse of CPs [6, 5], as future plans for COLORE include the incorporation of competency questions as requirements [10] to identify suitable ontologies.

References

1. van Benthem, Johan F. A. K. (1991) *The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, Springer; 2nd edition.
2. Bittner, T. (2004) Axioms for parthood and containment relations in bio-ontologies, *Proceedings of KR-MED 2004: First International Workshop on Formal Biomedical Knowledge Representation*.
3. Bittner, T. and Donnelly, M. (2005) Computational ontologies of parthood, componenthood, and containment. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence 2005 (IJCAI05). Kaelbling
4. Enderton, H.: Mathematical Introduction to Logic. Academic Press (1972)
5. Gangemi, A. (2005) Ontology Design Patterns for Semantic Web Content. *Proc. of International Semantic Web Conference*, pp. 262–267, Springer.
6. Gangemi, A., and Presutti, V. (2009) Ontology Design Patterns. In: *Handbook of Ontologies, 2nd Edition*, S. Staab (ed.), pp. 221–243, Springer.
7. Grüninger, M. (2009) Process Ontologies. In: *Handbook of Ontologies, 2nd Edition*, S. Staab (ed.), pp. 419–421, Springer.
8. Grüninger, M., Hahmann, T., Hashemi, A., and Ong, D. (2010) Ontology Verification with Repositories. *Proc. of the Sixth Int. Conference on Formal Ontologies in Information Systems (FOIS 2010)*, pp. 317–330, IOS Press.
9. Grüninger, M., Hahmann, T., Hashemi, A., Ong, D., and Ozgovde, A. (2012) Modular first-order ontologies via repositories, *Applied Ontology*, 7:169-210.
10. Katsumi, M. (2011) *A Methodology for the Development and Verification of Expressive Ontologies* (Master's Thesis). Retrieved from verb-https://tspace.library.utoronto.ca/bitstream/1807/31274/1/Katsumi_Megan_S_201111_MASc_thesis.pdf
11. Katsumi, M., and Grüninger, M. (2010) Theorem Proving in the Ontology Lifecycle. *Proc. of Knowledge Engineering and Ontology Design*
12. Ong, D. and Gruninger, M. (2011) Constructing an Ontology Repository: A Case Study with Theories of Time Intervals, Fourth International Workshop on Modular Ontologies. Ljubljana, Slovenia.
13. Scherp, A., Saathof, C., Franz, T. and Staab, S. (2011) Designing core ontologies, *Applied Ontology* 6:177-221.
14. Schorlemmer, M. and Kalfoglou, Y. (2008) Institutionalising Ontology-Based Semantic Integration, *Applied Ontology* 3:131-150.
15. Tsai, Hsing-chien (2009) Decidability of mereological theories, *Logic and Logical Philosophy* 18:45-63.