Formal Ontology in Information Systems
R. Ferrario and W. Kuhn (Eds.)

doi:10.3233/978-1-61499-660-6-9

What Is Ontology Reuse?

Megan KATSUMI
and Michael GRÜNINGER

Department of Mechanical & Industrial Engineering, Univerity of Toronto, Canada

Abstract. The reuse of ontologies is critical to their value as a means of knowledge representation. Unfortunately, reuse also still poses a considerable challenge for the ontological community. One reason for this is the lack of a formal definition of reuse. How can we attempt to perform or even assist this sort of ontology design, if we have no clear understanding of what constitutes reuse, and what does not? In this work we aim to remedy this situation by providing a formal definition of the concepts of reuse and reusability. Beyond providing a clear understanding of these concepts, part of the resulting definition is a characterization of the operations of reuse that can be leveraged to determine *how* a given ontology(s) must be reused to satisfy some specified requirements. This serves not only to provide direction for the task of reuse, but also to assess the implications of reusing an ontology(s), *a priori*. Collectively, the solutions presented in this paper serve as a major step in improving the current state of reuse.

Keywords. ontology design, development, reuse

1. Introduction

Reusability has long been recognized as a key attribute of ontologies, yet the principles and practice of reuse remain underdeveloped. The current lack of design through reuse presents a serious problem for the ontology community. While there are a variety of alternative approaches to knowledge representation, part of the case for ontologies is that they provide a knowledge representation that is not only *unambiguous* but *adaptable*, *shareable*, and *reusable*. Given that the reuse of ontologies remains such a challenging task [1,2], the benefits of shareability and reusability are often discounted.

Advocates of ontologies often claim that one benefit is that they are reusable. This is certainly true in principle, given that there are many generic concepts likely to appear (and thus be reused) in multiple applications. However, this is irrelevant if ontologies are not being reused in practice; if reuse is not commonplace we cannot claim that reusability is a benefit of ontologies. The value of ontologies is also motivated by the benefit of shareability. In general, this assumes that ontologies are developed with reuse. When an ontology is reused in the development of other ontologies, its semantics should be readily shared across their applications. If reuse does not occur, shareability may still be possible, but there will be no instances in which information can *readily* be shared between applications. Without these supposed benefits, the case for ontological solutions is less than convincing.

1.1. The Definition of Reuse: A Key Issue

Despite various attempts to aid and facilitate the reuse of ontologies, it remains a serious problem for the community. This is by no means an indication of ineffective or poor quality work, but a result of the difficulty of the problem. The scope of the problem of reuse is broad and poorly defined. In part, this is because reuse itself is currently not well-understood. There exist many approaches to reuse [1], formalized to various degrees, and yet there is no clear definition of what it means to develop an ontology via reuse, and precisely what this entails.

Not only does this lack of understanding inhibit the widespread practice of reuse, but where reuse *does* occur, the lack of a formal definition allows for the introduction of serious issues. Specifically, in the current state there are many instances where an ontology is said to be reused, but its semantics are completely changed. Situations such as this raise a key question: *What does it mean if one ontology is said to have reused another? What can we infer from this?*

1.2. Approach

We address these issues with several contributions. In Section 3 we derive a well-founded, formal definition for reuse from the notion of intended models. Through this definition, we define detailed reuse operations that completely characterize all possible cases of reuse. In Section 4, we discuss how it is possible to determine what reuse operations are required, and therefore provide explicit guidance and valuable insight for the task of reuse. The definition provides a concrete understanding of what it means to reuse an ontology (or ontologies), as well as the necessary concepts to describe reuse at an effective level of detail. This serves to improve both our understanding of reuse and our ability to support it.

2. Related Work

Currently, we find no formal definition of ontology reuse accepted or even proposed within the community. Nearly all work pertaining to the task of reuse assumes some implicit definition of the reuse of an ontology; no effort is made, formal or otherwise, to provide further clarification of precisely what this entails. An exception to this is found in the sort of definition presented by [3], where the authors describe reuse as:

> ...the process in which available (ontological) knowledge is used as input to generate new ontologies.

This is a very broad definition, and its lack of specificity not only limits its usefulness but, we will see later, is also inaccurate in its generality.

A similar, implicit definition may be found when reviewing the guidelines for reuse in [4], where the authors' customizing activity (Activity 2) accounts for a wide range of loosely defined operations (pruning, enriching, translating, and adapting the selected

[1] While there are certainly other aspects of reuse to consider, such as the search for and selection of an ontology to reuse, this work focuses solely on the task of reusing a particular ontology(s), (i.e. after it has been chosen).

ontology). Without any justification or precise definition of what these operations entail, it appears as though the authors also consider reuse to include any scenario in which an available ontology is used as an input in the development of a new one. Attempting to extract an implicit definition from other existing guidelines only reinforces that this vague definition is generally accepted by the community.

At best, we find that some more specific, related, and sometimes overlapping subtypes of reuse have been defined, such as merging and alignment [5], integration [6,7], modular or 'safe' reuse [8], and the application of ontology patterns [9]. Even with the provision of examples and guidelines, these implicit definitions remain either vague or isolated to a specific type of reuse. No existing work has provided a concrete understanding of precisely what is, and what is *not* ontology reuse.

3. Defining Reuse

In ontology design, we are attempting to create a set of axioms that captures some intended semantics of a set of concepts. Reuse is often conceptualized as a special case of design; intuitively, it refers to the task of taking some existing ontology(s) and manipulating it in some way in order to satisfy the design requirements. This observation is supported by existing, informal definitions of reuse.

In addition, there is an implicit condition which is often not stated as it is perhaps assumed as common sense – reuse can only be performed on a *reusable* ontology. To illustrate this with an example, consider the development of a finance ontology. It would be possible to take some existing anatomy ontology, and through some series of operations create the required finance ontology. However, if no remnants of the anatomy ontology are preserved in the finance ontology then we would not really want to consider such behaviour to be reuse. In fact, in such a case we may as well have developed the finance ontology from scratch. While the intuition of manipulating an existing ontology is certainly a necessary part of reuse, it is not sufficient to define reuse. In order to define reuse, we first consider the notion of reusability in more detail and produce a formal definition. Following this, we define the different possible reuse operations, by which an ontology may be manipulated. The combination of the classification of operations and the condition of reusability will form the definition of reuse that we present here.

It is important to note that the work that follows is restricted in that it does not consider any signature translations of the ontology to be reused; in other words, we assume that the theories to be reused are axiomatized in the required signature, where applicable. This assumption is made in order to simplify the presentation of the definition, and while this may seem to be an oversimplification it is in fact quite reasonable. If an ontology has been selected for reuse, then the developer must at least implicitly observe some mappings between its signature and those of the requirements. It is then simple enough to satisfy the assumption of a shared signature through the application of these mappings between the signatures of the candidate and required theories. A straightforward process to accomplish this has in fact been suggested in previous work [10]. The definition presented here also assumes that no language translations are required; in other words the reused ontology(s) are in the same logical language as the ontology being developed. While not ideal, this is the norm in the current state and therefore a reasonable assumption to make.

3.1. Reusability

A perspective on the goal of ontology design is that its aim is to develop a set of axioms that captures the intended models. It follows that to satisfy the requirements with existing theories, we should be reusing theories that *in some way* characterize one or more of the different domains that comprise the class of the intended models (recall, we denote these as $\mathfrak{M}^{intended}$). As discussed, a key distinction between reuse and traditional development is that with the concept of reuse there is an implicit *constraint* on the acceptable (re-) design of the axioms. Simply put, if we claim that some ontology(s) has been reused, we expect and in fact should require that *some* remnants of the original ontology remain in the resulting ontology. For any ontology T that is reused to satisfy $\mathfrak{M}^{intended}$, the models of T must characterize at least some of the models of some part of $\mathfrak{M}^{intended}$. We can restate this condition as saying that each model in $\mathfrak{M}^{intended}$ must map to a model of T, or a model of a *subtheory* [2] of T. Depending on the nature of the ontology to be reused, $\mathfrak{M}^{intended}$ may only map to some of the models of T as shown in Figure 1 (i.e. if T is weaker than the required ontology). Note that we denote the models of an ontology T as $Mod(T)$.

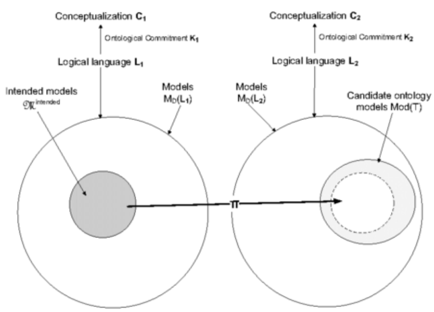

Figure 1. If the ontology is weaker than required, the intended models will only map to some of the models of its axiomatization. This and subsequent figures are adaptations of the depiction of intended models, originally from [11].

On the other hand, $\mathfrak{M}^{intended}$ may map to models outside of $Mod(T)$, i.e. models of some subtheory of T; this may occur if T is either stronger or incomparable to the required ontology, as shown in Figure 2 and Figure 3, respectively.

An additional factor to account for is that the class of intended models may characterize one or many different domains. For example, consider the design of an enterprise ontology: $\mathfrak{M}^{intended}$ will likely cover concepts of organizations, actors, dates, and so on. However, we do not necessarily expect to be able to reuse a single ontology that completely covers these enterprise-related concepts. More likely, we hope to find useful theories that contribute to the various required domains; we may reuse some ontology of time, another ontology of dates, and perhaps another ontology of organizations. In such cases, it is not all of $\mathfrak{M}^{intended}$ that maps in some way to T, but a *reduct* of $\mathfrak{M}^{intended}$ where the models are restricted so some sub-signature (e.g. only the time-related con-

[2]Note that we consider an ontology to be equivalent the logical closure of its set of axioms, i.e. a theory. We therefore use the term subtheory in the usual way in reference to an ontology (a theory), to refer to some weaker ontology, i.e. a 'sub-ontology'.

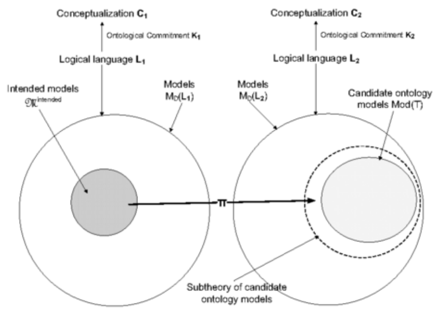

Figure 2. If the ontology is stronger than required, the intended models will map to models of a subtheory its axiomatization.

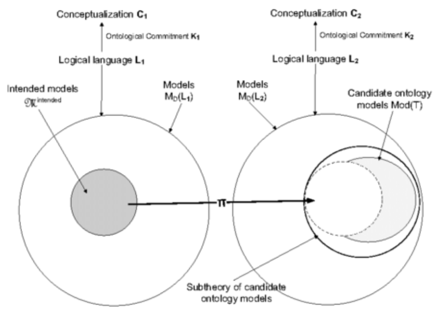

Figure 3. If the ontology is incomparable to the required ontology, the intended models will only map to some of the models of a subtheory of its axiomatization.

cepts). Similarly, there are cases where T may have a larger signature than that of the intended models. For example, in the design of our enterprise ontology we may reuse an ontology of time that is in fact part of a larger ontology for scheduling. In this case, the *reduct* of $\mathfrak{M}^{intended}$ maps to a *reduct* of the models of T. Therefore the diagrams that we have been considering do not necessarily capture mappings between the intended models and the models of the reused ontology, but the reducts of these models as shown more precisely in Figure 4. To account for mismatches in the scope of the required and reused ontology's concepts, we consider mappings between the *reducts* of the models. Thus L_1 may in fact be a sub-language (sub-signature) of the signature of the intended models, which we denote $\sigma(\mathfrak{M}^{intended})$, and similarly L_2 may be a sub-language (sub-signature) of the candidate's signature, which we denote $\sigma(T)$.

Here, we extend the usual meaning of a *reduct*, (denoted by $\mathcal{M}\mid_{\sigma}$) of a single model, $\mathcal{M}$, to some sub-signature, σ, of its original signature to apply to an entire class of models. Formally, we denote this $Red(\mathfrak{M},\sigma)$, and define it as follows:

Definition 1. *The reduct of a class of models $\mathfrak{M}$ to some signature σ is defined as the class of structures consisting of the reducts of each model $\mathcal{M}$ in $\mathfrak{M}$ to the signature σ:*
$Red(\mathfrak{M},\sigma) = \{\mathcal{N} : \mathcal{N} \cong \mathcal{M}\mid_{\sigma}, \mathcal{M} \in \mathfrak{M}\}$

Collecting all of these observations, we can refine our intuitions to say more specifically that if an ontology T is reusable to satisfy $\mathfrak{M}^{intended}$, then there must be a mapping

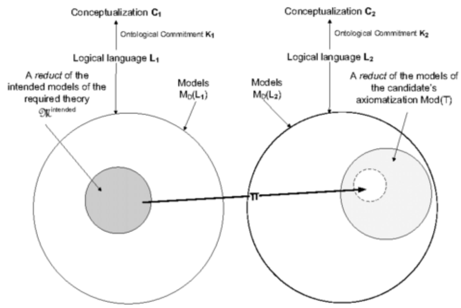

Figure 4. Intuitively, if an ontology is reusable for some required ontology, we expect to find some part of it in the resulting ontology.

from the reducts of $\mathfrak{M}^{intended}$ to reducts of models of T', a subtheory of T with the same signature. Formally,

Definition 2. *T is reusable for $\mathfrak{M}^{intended}$ iff:*
there is a mapping $\pi : Red(\mathfrak{M}^{intended}, \sigma_1) \rightarrow Red(Mod(T'), \sigma_2)$ where:

- $T' \subseteq T$,
- $\sigma(T') = \sigma(T)$
- $\sigma_1 \subseteq \sigma(\mathfrak{M}^{intended})$
- $\sigma_2 \subseteq \sigma(T')$

It is straightforward to extend this definition to apply to a collection of ontologies; the same intuition applies:

Definition 3. *For a set of ontologies, $T_1, ...T_n$, we say that the set $T_1, ...T_n$ is reusable for $\mathfrak{M}^{intended}$ iff each T_i is reusable for $\mathfrak{M}^{intended}$.*

This notion of reusability can be captured similarly, from the perspective of the theories' axiomatizations. Note that owing to our definition of an ontology being the logical *closure* of a set of axioms, the $\subseteq$ symbol denotes a *subtheory* as opposed to simply a subset of axioms. We use $Th(\mathfrak{M})$ to denote the axiomatization of a class of models $\mathfrak{M}$.

Theorem 1. *A set of ontologies $T_1, ...T_n$ is reusable for $\mathfrak{M}^{intended}$ iff $Th(\mathfrak{M}^{intended})$ contains non-trivial subtheories of $T_1, ..., T_n$.* [3]

3.2. Classification of Reuse Operations

While it is tempting to interpret Theorem 1 as a definition of *reuse*, it would be inaccurate to do so. Certainly, for any ontology that has been reused, we expect that it must have been reusable and thus we expect the results of reuse are captured by Theorem 1. In fact, the theorem captures the basic intuition of reuse that motivated the definition of reusability: in order for Design to qualify as reuse, we expect some remnants of the original ontology(s) to remain. However, reusability is a necessary but not sufficient condi-

[3]The proof for this and all other theorems, as well as an extended discussion of examples, may be found at: `http://stl.mie.utoronto.ca/publications/MeganKatsumi_PhD_Thesis.pdf`.

tion for reuse. There is an extralogical condition that must be accounted for in order to completely capture what it means to reuse an ontology. Unlike reusability, reuse is not a static property between theories; reuse refers to an act that is performed with some existing theories, in the design of an ontology. Simply because an ontology is reusable for some intended models does not mean it will or has been be reused to axiomatize those models.

Informally, reuse is the act of applying some operations to a given, reusable ontology(s), such that the final result axiomatizes the intended models. To formally define reuse, these operations must be completely identified and defined. While various approaches to reuse, such as ontology fusion, merging, and extension, have been identified in varying degrees of detail in the current state, none of these approaches have been defined with respect to a complete definition of reuse; in some cases they have not been defined formally at all. Here, we provide a precise definition for a set of operations that completely covers the possible approaches to reuse. The terms used here should be interpreted independently of those that have been identified in the literature. No relationships should be inferred or assumed due to a similarity of terms or descriptions.

The following are the 4 distinct *reuse operations* by which an ontology may be manipulated for reuse: as_is, extraction, extension, and combination. These operations are natural and fairly obvious approaches to manipulating an ontology; the focus here is on the way in which we define them, and the subsequent analysis they are capable of supporting. With the exception of as is, for each of these operations, we identify three more precise, *specialized operations* based on a more precise identification of the changes to the original ontology.

As is refers to a sort of null operation. This corresponds to the reuse of an ontology directly, with no modifications of any sort. In this sense, it is analogous to an identity function and we consider it to be a *trivial* operation. It can be formally defined as follows:

Definition 4. $\mathit{as\_is}(T) = T$

Extraction refers to the reuse of an ontology via a removal of some of the original axioms, denoted by T^-. It can be formally defined as follows[4]:

Definition 5. $\mathit{extraction}(T, T^-) = T/T^-$

We identify the following three specializations of the extraction operation:

Domain Extraction: an entire domain (set of concepts) is completely extracted from the original ontology. Observe that T conservatively extends domain_extraction(T/T^-).

Definition 6. $\mathit{domain\_extraction}(T, T^-) = T/T^-$ *where* $\sigma(T^-) \cap \sigma(T/T^-) = \emptyset$

Weakening Extraction: the semantics of the original ontology are weakened by the operation while its scope remains the same. Observe that T non-conservatively extends weakening_extraction(T, T^-).

Definition 7. $\mathit{weakening\_extraction}(T, T^-) = T/T^-$ *where* $\sigma(T/T^-) = \sigma(T)$

[4]The / symbol denotes the set difference between two theories.

Weakening Domain Extraction: the semantics of the original ontology are weakened and only some of the concepts are reused with this operation. No part of this extraction may be comprised of a weakening_extraction or a domain_extraction; this restriction is captured by the last condition in the definition which simply states that there are no subtheories of the extraction that could be extracted from T as a weakening_extraction or a domain_extraction. Observe that T non-conservatively extends weakening_domain_extraction(T, T^-).

Definition 8. *weakening_domain_extraction*$(T, T^-) = T/T^-$ *where* $\sigma(T/T^-) \subset \sigma(T)$ *and there does not exist a* $T^-_{sub} \subset T^-$ *such that* $\sigma(T^-_{sub}) \cap \sigma(T/T^-_{sub}) = \emptyset$ *or* $\sigma(T/T^-_{sub}) = \sigma(T)$

The first two operations addressed the sort of do-nothing operation, and the operation to remove axioms from an ontology. Now we consider the operations to add axioms to an ontology; the way that this occurs depends on the source of the axioms – they could be new axioms, created during design by the ontology developer, or they could be existing axioms, reused from some other ontology. It is important to make this distinction because these differences affect the way in which reuse is carried out. If the axioms were reused from another ontology, the design work is minimal, however if they are new axioms then the developer must have invested some time to design them from-scratch. The distinction between these two types of axiom addition also has potential implications for the perception of the resulting shareability. If the additional axioms were reused from some other ontology it *may* indicate that shareability will also be supported with this ontology; at the very least it indicates that shareability is something that should be considered and addressed in the metadata. We formalize this distinction by considering whether the additional axioms may be found in some repository. If axioms are added to one ontology (by reuse of axioms) from another ontology, then a *combination* has occurred. Otherwise, the addition is simply an *extension* of the ontology with additional axioms. We make reference to a single repository for simplicity, however it is straightforward to see that the definition and subsequent results also apply for any number of repositories or other sources of ontologies.

Extension refers to the reuse of an ontology via the creation and introduction of new axioms, denoted by T^+. It can be formally defined as follows:

Definition 9. *Let* $\mathbb{S}$ *be some ontology repository.*
extension$(T, T^+) = T \cup T^+$ *where* $T^+ \notin \mathbb{S}$

We identify the following three specializations of the extension operation:

Domain Extension: the original ontology is extended via a new set of axioms created by the developer, T^+, in a completely new (distinct) domain. Observe that domain_extension(T, T^+) conservatively extends T.

Definition 10. *Let* $\mathbb{S}$ *be some ontology repository.*
domain_extension$(T, T^+) = T \cup T^+$ *where* $T^+ \notin \mathbb{S}$ *and* $\sigma(T) \cap \sigma(T^+) = \emptyset$

Strengthening Extension: the original ontology is extended via a new set of axioms created by the developer, T^+, such that its semantics are strengthened while maintaining its scope. Observe that strengthening_extension(T, T^+) non-conservatively extends T.

Definition 11. *Let $\mathbb{S}$ be some ontology repository.*
$\mathit{strengthening\_extension}(T,T^+) = T \cup T^+$ *where* $T^+ \notin \mathbb{S}$ *and* $\sigma(T^+) \subseteq \sigma(T)$

Strengthening Domain Extension: the original ontology is extended via a new set of axioms created by the developer, T^+, that both strengthens its original concepts and adds to them, thereby expanding the scope of its domain. No part of this extension may be comprised of a domain_extension or a strengthening_extension.Observe that strengthening_domain_extension(T,T^+) non-conservatively extends T.

Definition 12. *Let $\mathbb{S}$ be some ontology repository.*
$\mathit{strengthening\_domain\_extension}(T,T^+) = T \cup T^+$ *where the following three conditions hold:*

1. $T^+ \notin \mathbb{S}$;
2. $\sigma(T) \subset \sigma(T^+)$, *or*
   $\sigma(T) \cap \sigma(T^+) \neq \emptyset$, $\sigma(T) \not\subseteq \sigma(T^+)$, $\sigma(T) \not\supseteq \sigma(T^+)$ *(i.e. signatures overlap);*
3. *there does not exist a* $T^+_{sub} \subset T^+$ *such that* $\sigma(T^+_{sub}) \subseteq \sigma(T)$ *or* $\sigma(T^+_{sub}) \cap \sigma(T) = \emptyset$

Combination refers to the reuse of an ontology via the addition of other reused ontology(s), denoted T_2. It can be formally defined as follows:

Definition 13. *Let $\mathbb{S}$ be some ontology repository.* $\mathit{combination}(T_1,T_2) = T_1 \cup T_2$ *where* $T_2 \in \mathbb{S}$

We identify the following three specializations of the combination operation:

Domain Combination: The original ontology, T is combined with another ontology T_2 that defines a completely distinct domain. Observe that domain_combination(T,T_2) conservatively extends T (and T_2).

Definition 14. *Let $\mathbb{S}$ be some ontology repository.* $\mathit{domain\_combination}(T,T_2) = T \cup T_2$ *where* $T_2 \in \mathbb{S}$ *and* $\sigma(T) \cap \sigma(T_2) = \emptyset$

Strengthening Combination: The original ontology, T, is combined with another ontology that defines the same domain, T_2, such that their semantics are strengthened while maintaining the original scope. Observe that strengthening_combination(T,T_2) non-conservatively extends T (and T_2).

Definition 15. *Let $\mathbb{S}$ be some ontology repository.*
$\mathit{strengthening\_combination}(T,T_2) = T \cup T_2$ *where* $T_2 \in \mathbb{S}$ *and* $\sigma(T_2) \subseteq \sigma(T)$

Strengthening Domain Combination: The original ontology, T, is combined with another ontology T_2 that strengthens the concepts of T while also introducing new ones. No part of this extension may be comprised of a domain_extension or a strengthening_extension.Observe that strengthening_domain_combination(T,T_2) non-conservatively extends T (and T_2).

Definition 16. *Let $\mathbb{S}$ be some ontology repository.*
$\mathit{strengthening\_domain\_combination}(T,T_2) = T \cup T_2$ *where the following three conditions hold:*

1. $T_2 \in \mathbb{S}$;
2. $\sigma(T) \subset \sigma(T_2)$, *or*
   $\sigma(T) \cap \sigma(T_2) \neq \emptyset$, $\sigma(T) \not\subseteq \sigma(T_2)$, $\sigma(T) \not\supseteq \sigma(T_2)$ *(i.e. signatures overlap);*
3. *there does not exist a* $T_2' \subset T_2$ *such that* $\sigma(T_2') \subseteq \sigma(T)$ *or* $\sigma(T_2') \cap \sigma(T) = \emptyset$

While it is intuitive to consider reuse as the application of a *sequence* of these operations, so long as an ontology or set of ontologies $T_1, ...T_n$ is reusable for $Th(\mathfrak{M}^{intended})$ the *order* in which reuse operations are applied does not affect the final result. The order may be of some importance in maintaining consistency throughout the series of operations, however this is not a requirement of design. [5]

Again, consider the definition of reuse – this time based on the reuse operations we've just defined. Such a definition appeals to our intuition: *reuse is the application of some set of reuse operations*; however the act of reuse cannot be defined via these operations alone. As discussed, it is not the case that *any* set of these operations corresponds to an act of reuse. Reuse is defined as the act of performing some set of operations on some existing ontology(s) $T_1, ..., T_n$ [6] that is reusable for some $\mathfrak{M}^{intended}$, in order to transform $T_1, ..., T_n$ to $Th(\mathfrak{M}^{intended})$. Formally,

Definition 17. *$T_1, ..., T_n$ are reused for $\mathfrak{M}^{intended}$ iff*

- *$T_1, ..., T_n$ are reusable for $\mathfrak{M}^{intended}$, and*
- *some set of reuse operations applied to $T_1, ..., T_n$ axiomatizes $\mathfrak{M}^{intended}$*

The section that follows explores the ramifications of this result and the transparency it provides for the task of reuse.

4. Implications

The definitions presented in the previous section fill a critical void for the task of ontology reuse. In order for ontology development to move toward becoming an engineering discipline, concepts like reuse and its operations must be clearly understood and defined. Here, we consider the implications of the definitions presented in the previous section; in particular, the improved understanding and the opportunities for reuse support that they provide.

4.1. Validation of Reuse Operations

The reuse operations defined not only help to identify the sorts of manipulations that may be performed, they provide a complete characterization of the possible ways in which an ontology(s) may be reused; through this they achieve an explicit understanding of what is, and what is not reuse.

Theorem 2. *Let $\mathbb{S}$ be some ontology repository, and let $T_1, ..., T_n$ be some ontologies in $\mathbb{S}$. If an ontology or set of ontologies $T_1, ...T_n$ is reusable for $\mathfrak{M}^{intended}$ there exists a set of specialized reuse operations on $T_1, ..., T_n$ to axiomatize $\mathfrak{M}^{intended}$.*

---

[5]The proof of this commutativity between reuse operations is also included in the referenced thesis.

[6]Note that we use the term ontology in a broad sense – this work includes the possibility that $T_1, ..., T_n$ are some combination not only of whole ontologies being reused, but perhaps modules of ontologies, an upper ontology(s), or even ontology patterns.

Proof Sketch Consider the reuse of a single theory T for $\mathfrak{M}^{intended}$. There are 4 possible relationships between T and $Th(\mathfrak{M}^{intended})$. For each relationship, the necessary reuse operation to axiomatize $Th(\mathfrak{M}^{intended})$ follows by definition of the operations:

- $T \models Th(\mathfrak{M}^{intended}) \rightarrow \mathsf{extraction}(T, T^-) = Th(\mathfrak{M}^{intended})$ for some T^-
- $Th(\mathfrak{M}^{intended}) \models T \rightarrow \mathsf{extension}(T, T^+) = Th(\mathfrak{M}^{intended})$ for some T^+
- $T = Th(\mathfrak{M}^{intended}) \rightarrow \mathsf{as\_is}(T)$
- $T \parallel Th(\mathfrak{M}^{intended}) \rightarrow \mathsf{extension}(\mathsf{extraction}(T, T^-), T^+)$ for some T^-, T^+ where the $\parallel$ symbol denotes that two theories are incomparable

The extension of this result for multiple theories $T_1, ...T_n$ is straightforward and relies on the commutativity of reuse operations; by combining the theories we can reduce this case to the reuse of a single theory. □

Theorem 2 confirms that our definition of reuse operations has in fact covered all cases of reuse.

4.2. *Guiding Ontology Reuse*

Not only does there exist some set of operations to axiomatize the intended models, we can in fact leverage the definitions to determine precisely what operations are required. This result comes about by way of a constructive proof of Theorem 2. As outlined in the proof sketch presented in the previous section, the approach relies on assessing the relationship between $Th(\mathfrak{M}^{intended})$ and the ontology(s) to be reused. Assuming that T is reusable for $\mathfrak{M}^{intended}$, we can completely characterize all possible reuse cases based on the relationship between the signature and the axioms of $Th(\mathfrak{M}^{intended})$ and T. This result can also be extended to identify the specialized reuse operations required, and follows directly from the full proof of Theorem 2. Thus for any ontology, either *a priori* or post-reuse, the developer may assess what reuse operations must be or were applied. This is valuable in both providing guidance for reuse, and also in assessing precisely *how* a particular ontology was reused.

In practice, $Th(\mathfrak{M}^{intended})$ will likely not be known completely and thus some approximation (such as the set of competency questions) will be substituted. While the use of competency question (CQs) as requirements is well-established, their role as approximating the required ontology, $Th(\mathfrak{M}^{intended})$, may be subject to speculation. We defend this approach from a pragmatic standpoint; at this stage in development, no better approximation of the ontology exists. Certainly, there is a difficulty that the greater the precision of the CQs, the closer the developer is to developing the ontology from-scratch, however this challenge exists for the task of requirements specification across all sorts of disciplines: a compromise between precision and practicality must be found. The use of an approximation of the required ontology is not something that is unique to this approach to reuse, it is a norm of development methodologies in general and thus should not be regarded as a weakness of this approach.

4.3. *Assessing Shareability*

Extraction, extension and combination may be performed in several distinct ways, each of which has a different intuition and impact on the resulting ontology. The specializations of the reuse operations are based on a more precise identification of these changes to the original ontology. Not only does this provide more detailed guidance with respect to the required operations, it allows for a more detailed analysis of the implications of any given reuse scenario. The identification of these operations facilitates recognition of the relationships between concepts in the original ontology(s) and the resulting ontology being developed via reuse. Most notably, this supports the identification of shareability that is attained or lost for concepts in the new ontology.

The identification of necessary specialized operations may also be used as a sort of look-ahead in order to inform developers of the implications that reuse of a particular ontology will have on its semantics, and consequently the resulting shareability that can be expected. In the case that multiple reuse operations are necessary, the assessment of the resulting shareability follows easily: any operation that does not preserve the semantics of T implies that overall, its semantics are not preserved. The assessment on the resulting semantics is similar for cases where multiple ontologies are reused. The result serves to inform the developer of the shareability between the resulting ontology and the reused theories. In effect, it enables us to give a useful answer the question: *What does it mean if one ontology is said to have reused another?*

4.4. Other Operations on Ontologies

The reader may note that the notion of ontology operations itself is not novel. Work on ontology algebras, belief revision, and the Distributed Ontology Language (DOL) standard have particular resonance. We review each of these areas here and discuss the relationship to this work. It is critical to note that the novelty of the current work is not in the identification of the operations themselves, rather, the contribution we make is the way in which we interpret these operations, *in the context of a formal definition of reuse*. The perspective we take in defining these operations is such that they can serve to both prescribe how an ontology can be reused to satisfy a given set of requirements, and to assess what the implications of this reuse will be on the resulting ontology.

4.4.1. Ontology Algebras

The ontology algebras presented in [12,13] are designed to support ontology integration, thus the resulting operations focus on composing ontologies when global consistency is not feasible. We observe that similarities are evident between the operators defined in this work and the reuse operations we identify here. In particular, the combination and extraction operations may be captured by operators defined for ontology algebras. A major distinction here is that the role of the algebra operators is to support the integration ontologies, thus their attention is on the combinations of ontologies. The intersection and difference operators defined for the ontology algebra do not correspond to any of the reuse operations we define here, as these distinctions are not relevant for our purposes; nor do the algebras define any sorts of extension operations, as their focus is on the combination of existing theories, not the addition of new axioms.

4.4.2. Belief Revision and the AGM Framework

The reader may also have noticed similarities between the identified operations and the well-known AGM Framework for belief revision [14,15].It defines operations, or ways in which beliefs can change. The notion of revising a set of beliefs to resolve inconsistencies is thus one of particular importance. In the context of ontologies, the AGM Framework may be considered as a tool to approach ontology *evolution* [16].

Conceptually, the extraction, extension, and combination operations could all be described in the AGM Framework, albeit without the sorts of distinctions that are made between the specializations. On the other hand, there are belief revision operations such as *revision* and *consolidation* do not correspond to any of the reuse operations defined here. Again, a major distinction is that the purpose of these operations is to update the belief set and arrive at a consistent set of beliefs, whereas the reuse operations are manipulating existing ontology(s) in order to arrive at a specific end result.

4.4.3. DOL

The Distributed Ontology Language (DOL) [17], has been designed in response to the OntoIOP request for proposal [18] to address the challenge of interoperability for heterogeneous formal representations. The DOL project considers not only ontologies, but 'specifications and MDE [Model-Driven Engineering] models' in a variety of languages, and is focused not on reuse but on integration and interoperability. While the scope of this project is much broader and not quite aligned with the work presented here, we find that the 'structuring language for OMS' provided by DOL corresponds in a way to the concept of reuse operations. A key distinction is that the language in DOL defines constructs, which are meant to describe the structure of theories and how they relate to one another, as opposed to the definition of operations here which represents a manipulation *performed* on a given ontology.

Each of the reuse operations defined here may be expressed by some DOL construct. However certain distinctions such as the difference between the original and resulting signature that are made when considering operations in the context of reuse, are not captured by these constructs as they are simply not relevant when defining metalogical relationships between theories. As mentioned, the DOL constructs provide a means of describing the structure of an ontology such that we can understand its relationship to other ontologies. The concept of an 'original' (reused) ontology, or even the source of axioms (another ontology, or created by the developer) make little sense from this perspective.

5. Conclusion

The definition of reuse presented here is not only novel, but the first of its kind. No definition of reuse or complete characterization of operations has been provided for the community before. While there are certainly other challenges that present barriers for the reuse of ontologies, the work presented in this paper serves as a substantial contribution towards the goal of reuse becoming a more principled and effective means of ontology design. We developed definitions for the concepts of reuse and reusability that are not only formal but functional. We presented a precise set of reuse operations that, *in any possible case*, could be applied to an ontology in order to satisfy some specified requirements. Not only are the operations capable of providing valuable guidance for reuse, perhaps more crucially they provide a means of assessing and understanding the implications of a given instance of reuse.

Of particular interest for the task of ontology reuse are the relationships identified between the reuse operations and the constructs specified in DOL. The correspondences that we have identified present an opportunity to extend implementation of this work to capture any resulting ontologies in the DOL terminology, which is an OMG adopted specification. For example, it will be straightforward to translate the necessary reuse operations to axiomatize the intended models from some $T_1, ... T_n$, in order to define the structure of the resulting ontology using the metalogical relationships of DOL. This has the potential to ease adoption of the DOL standard, and also to benefit reuse with the provision of standardized metadata detailing the reuse of ontologies.

We hope that future work will look towards the adoption of the approach to reuse presented here in a more user-friendly medium, perhaps integrated with existing ontology design tools. While this work provides the necessary theory to support reuse, tool support to implement this definition would be an invaluable contribution towards simplifying and improving the practice of reuse.

References

[1] C. K. Janowicz, O. Kutz, C. Lange, A. Levenchuk, F. Quattri, A. Rector, T. Schneider, S. Spero, A. Thessen, M. Vegetti, et al. Semantic web and big data meets applied ontology—The ontology summit 2014. *To appear in: Applied Ontology*, 2014.

[2] E. Simperl. Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering*, 68(10):905–925, 2009.

[3] E. P. Bontas, M. Mochol, and R. Tolksdorf. Case studies on ontology reuse. In *Proceedings of the IKNOW05 International Conference on Knowledge Management*, volume 74, 2005.

[4] M. Fernández-López, M. C. Suárez-Figueroa, and A. Gómez-Pérez. Ontology development by reuse. In *Ontology Engineering in a Networked World*, pages 147–170. Springer, 2012.

[5] N. F. Noy and M. A. Musen. An algorithm for merging and aligning ontologies: Automation and tool support. In *Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 1999–0799, 1999.

[6] H. S. Pinto and J. P. Martins. A methodology for ontology integration. In *Proceedings of the 1st international conference on Knowledge capture*, pages 131–138. ACM, 2001.

[7] A. Gangemi, D. Pisanelli, and G. Steve. Ontology integration: Experiences with medical terminologies. In *Formal ontology in information systems*, volume 46, pages 98–94. IOS Press, Amsterdam, AM, 1998.

[8] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.(JAIR)*, 31:273–318, 2008.

[9] R. A. Falbo, G. Guizzardi, A. Gangemi, and V. Presutti. Ontology patterns: clarifying concepts and terminology. In *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188*, pages 14–26. CEUR-WS. org, 2013.

[10] M. Katsumi and M. Grüninger. Choosing ontologies for reuse. *Submitted to: Applied Ontology*, 2016.

[11] N. Guarino, D. Oberle, and S. Staab. *What is an Ontology?*, pages 1–17. Springer-Verlag, Berlin, 2 edition, 2009.

[12] P. Mitra and G. Wiederhold. An ontology-composition algebra. In *Handbook on ontologies*, pages 93–113. Springer, 2004.

[13] G. Wiederhold. An algebra for ontology composition. In *Proceedings of 1994 Monterey Workshop on Formal Methods*, volume 56, page 61. Citeseer, 1994.

[14] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The journal of symbolic logic*, 50(02):510–530, 1985.

[15] P. Gärdenfors. *Belief revision*, volume 29. Cambridge University Press, 2003.

[16] G. Flouris, D. Plexousakis, and G. Antoniou. Evolving ontology evolution. In *SOFSEM 2006: Theory and Practice of Computer Science*, pages 14–29. Springer, 2006.

[17] A. Hoffmann, L. Obrst, E. Kendall, T. Athan, and T. Mossakowski. The distributed ontology, modeling, and specification language (DOL). Technical report, OMG, 2015.

[18] D. F. Neuhaus. Ontology, model and specification integration and interoperability (OntoIOp) RFP. Technical report, OMG, 2013.