A Semantic Importing Approach to Knowledge Reuse from Multiple Ontologies

Jie Bao, Giora Slutzki and Vasant Honavar
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1040, USA
{baojie,slutzki,honavar}@cs.iastate.edu

Abstract

We present the syntax and semantics of a modular ontology language $\mathcal{SHOIQP}$ to support context-specific reuse of knowledge from multiple ontologies. A $\mathcal{SHOIQP}$ ontology consists of multiple ontology modules (each of which can be viewed as a $\mathcal{SHOIQ}$ ontology) and concept, role and nominal names can be shared by "importing" relations among modules. $\mathcal{SHOIQP}$ supports contextualized interpretation, i.e., interpretation from the *point of view* of a specific package. We establish the necessary and sufficient constraints on domain relations (i.e., the relations between individuals in different local domains) to preserve the satisfiability of concept formulae, monotonicity of inference, and transitive reuse of knowledge.

1 Introduction

The success of the world wide web can be attributed to the *network effect*: The absence of central control on content and organization of the web allows thousands of independent actors to contribute resources (web pages) that are interlinked to constitute the web. Recent efforts to extend the web into a *semantic web* are aimed at enriching the web with machine interpretable content and interoperable resources and services. Realizing the full potential of the semantic web requires the large-scale adoption and use of ontology-based approaches to sharing of information and resources. In such a setting, instead of a single, centralized ontology, it is much more natural to have multiple distributed ontologies that cover different, perhaps partially overlapping, domains (e.g., biology, medicine, pharmacology). Such ontologies represent the *local* knowledge of the ontology designers, that is, knowledge that is applicable within a specific *context*. Hence, there is an urgent need for theoretically sound yet practical approaches that support user, context, or application-specific adaptation and reuse of knowledge from multiple autonomously developed ontologies in specific applications.

Ontologies on the semantic web need to satisfy two apparently conflicting objectives (Bouquet *et al.* 2003): *Sharing* or *r*euse of knowledge across autonomously developed ontologies; and accommodation of the *local points of view* or *contextuality* of knowledge. Consequently, there have been several efforts aimed at developing formalisms that allow reuse of knowledge from multiple ontologies via *contextualized interpretations* in multiple local domains instead of a single shared global interpretation domain. Contextualized reuse of knowledge requires the interactions between local interpretations to be controlled. Examples of such modular ontology languages include: Distributed Description Logics (DDL) (Borgida & Serafini 2003), $\mathcal{E}$-Connections (Grau, Parsia, & Sirin 2004), Package-based Description Logics (P-DL) (Bao, Caragea, & Honavar 2006b) and Semantic Importing (Pan, Serafini, & Zhao 2006).

An alternative approach to knowledge reuse is based on the notion of *conservative extension* (Ghilardi, Lutz, & Wolter 2006; Grau *et al.* 2007; 2006; Grau & Kutz 2007) which allows ontology modules to be interpreted using standard semantics by requiring that they share the same global interpretation domain. To avoid undesired combination of ontology modules, this approach requires the combination of ontology modules to be a conservative extension of component modules. More precisely, if O is the union of a set of ontology modules $\{O_1, ..., O_n\}$, then we say O is a conservative extension of O_i if $O \models \alpha_i \Leftrightarrow O_i \models \alpha_i$ for any α_i of the form $C_1 \sqsubseteq C_2$, where C_1, C_2 are concepts in the language of O_i. This guarantees that combining knowledge from several ontology modules does not alter the consequences of knowledge contained in any component module. Thus, a combination of ontology modules cannot induce a new concept inclusion relation between existing concepts in any of the component modules. This requirement is enforced through a syntactical restriction that forbids the use of any axiom that is not "local" to an ontology module (e.g., $\top \sqsubseteq C$).

Current approaches to knowledge reuse have several limitations. To preserve contextuality, existing modular ontology languages offer only limited ways to interconnect ontology modules (and hence limited ability to reuse knowledge across modules). For instance, DDL does not allow concept construction using foreign roles or concepts; $\mathcal{E}$-Connections does not allow concept inclusion between ontology modules or the use of foreign roles; P-DL and Semantic Importing in their current forms require each component module to be in $\mathcal{ALC}$. None of the existing approaches support knowledge reuse in a setting where each ontology module uses a representation language that is as expressive as OWL-DL, i.e.

$\mathcal{SHOIN}(D)$.

Furthermore, some of the existing modular ontology languages suffer from reasoning difficulties that can be traced to an absence of natural ways to restrict the relations between individuals in different local domains. For example, DDL does not support the transitivity of inter-module concept subsumptions (known as *bridge rules* in DDL) in general, and a concept that is declared as being more specific than two disjoint concepts in another module may still be satisfiable (the inter-module satisfiability problem) (Bao, Caragea, & Honavar 2006b; Grau, Parsia, & Sirin 2004). Unrestricted use of generalized links in $\mathcal{E}$-Connections has also been shown to lead to reasoning difficulties (Bao, Caragea, & Honavar 2006a).

Conservative extensions (Grau *et al.* 2007; 2006; Grau & Kutz 2007) in their current form, since they require a single global interpretation domain, do not allow different modules from interpreting axioms within their own local contexts. Hence, the designers of different ontology modules have to anticipate all possible contexts in which knowledge from a specific module might be reused. Thus, locality of knowledge is ensured by precluding several modeling scenarios that would otherwise be quite useful in practice, e.g., the refinement of relations between existing concepts in an ontology module, and the reuse of nominals (Lutz, Walther, & Wolter 2007).

Against this background, this paper explores a formalism that can support *context-aware reuse* from multiple ontology modules. The resulting modular ontology language:

- Allows each ontology module to use subset of $\mathcal{SHOIQ}$ (Horrocks & Sattler 2005), i.e., $\mathcal{ALC}$ augmented with transitive roles, role inclusion, role inversion, qualified number restriction and nominal concepts, hence covers a significant fragment of OWL-DL.

- Supports more flexible modeling scenarios than those supported by existing approaches, using a mechanism of *semantic* importing of names (including concept, role and nominal names) across ontology modules.

- Contextualizes the interpretation of reused knowledge. Locality of axioms in ontology modules is obtained "for free" by its *contextualized semantics*, thereby freeing ontology designer from the burden of ensuring the reusability of an ontology module in contexts that are hard to foresee at the time of construction of the module in question. A natural consequence of contextualized interpretation is that inferences that are drawn are always *from the point of view* of a *witness* module. Thus, different modules might infer different consequences, based on the knowledge that they import from other modules.

- Ensures that the result of reasoning is always the same as that obtained from a standard reasoner over an integrated ontology resulting from combining the relevant knowledge in a context-specific manner. This ensures the *monotonicity* of inference in the distributed setting.

- Avoids many of the reasoning difficulties of the existing approaches.

2 Semantic Importing

This section introduces the syntax and semantics of the proposed language.

2.1 Syntax

Definition 1 *A distributed TBox contains a set of modules called* packages*, each is a TBox of a subset of $\mathcal{SHOIQ}$. Each package P has associated with it, a local signature: a subset of its symbols (the set of concept, role and individual names)* $\mathsf{Loc}(P) \subseteq \mathsf{Sig}(P)$*; for any symbol $s \in \mathsf{Loc}(P)$, P is the* home package *of s, denoted by $P = \mathsf{Home}(s)$. The set of symbols in $\mathsf{Imp}(P) = \mathsf{Sig}(P) \backslash \mathsf{Loc}(P)$ is called P's* imported signature.

If a symbol $s \in \mathsf{Loc}(Q)$ also appears in the imported signature of a different package P (i.e., $s \in \mathsf{Imp}(P)$), we say that P imports *$Q : s$ and denote it as $Q \xrightarrow{s} P$. If any local symbol of Q is imported into P, we say that P imports Q and denote it as $Q \mapsto P$.*

Each package P_i has an associated context which constrains the scope of knowledge in it. In particular, for each package P_i, instead of the universal top ($\top$) concept, we have its contextualized counterpart $\top_i$, and (global) negation ($\neg$) is replaced by its contextualized counterpart negation $\neg_i$.

A formula (i.e., a concept, role, nominal or general concept inclusion (GCI) axiom) is called a pure i-formula if each of the names used in the formula is in $\mathsf{Loc}(P_i) \cup \{\top_i\}$. We refer to an formula in a package P_i that uses names from local signatures of packages other than i a hybrid i-formula. An i-formula can be either a pure i-formula or a hybrid i-formula. We denote an i-formula X by $i : X$, and we drop the prefix $i :$ when there is no possibility of confusion.

Without loss of generality, we assume $\neg_i$ can be applied only to an i-concept[1]*. P_j can use $\neg_i$ only when $P_i \mapsto P_j$.*

The importing transitive closure *$I(P_i)$ of a package P_i contains packages that are directly or indirectly imported by P_i. That is,*

- $\forall j \neq i, P_j \mapsto P_i \Rightarrow P_j \in I(P_i)$
- $\forall k \neq j \neq i, (P_k \mapsto P_j) \wedge (P_j \in I(P_i)) \rightarrow P_k \in I(P_i)$

We use P_i^* to denote the union of a package P_i and its importing transitive closure $I(P_i)$ and importing relations among them.

A distributed TBox $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j} \rangle$ has acyclic importing relation *if for any $i \neq j$, $P_j \in I(P_i) \rightarrow P_i \notin I(P_j)$, otherwise it has* cyclic importing relation. *Σ is* closed *if every symbol used in Σ is defined in one of its component packages, i.e., for $\forall s, \forall k, s \in \mathsf{Imp}(P_k) \rightarrow \mathsf{Home}(s) \in \{P_i\}$.*

We denote a package-based Description Logics (DL) by adding the letter $\mathcal{P}$ to the notation for the corresponding DL. Thus, $\mathcal{ALCP}$ is the package-based DL $\mathcal{ALC}$. In this paper we focus on $\mathcal{SHOIQP}$ thereby extending some of the results of (Bao, Caragea, & Honavar 2006c) which studied

[1] For an i-concept C, we can always transform $\neg_j C$ appearing in package k to $\neg_j C'$ where C' is a new j-concept and add an axiom $C' = C$ in package k, k and j may or may not be the same.

$$
\begin{array}{rcl}
R^{\mathcal{I}_i} & = & (R^{\mathcal{I}_i})^+, \text{for transitive role role } R \\
(R^-)^{\mathcal{I}_i} & = & \{\langle x, y\rangle | \langle y, x\rangle \in R^{\mathcal{I}_i}\} \\
(C \sqcap D)^{\mathcal{I}_i} & = & C^{\mathcal{I}_i} \cap D^{\mathcal{I}_i}, \\
(C \sqcup D)^{\mathcal{I}_i} & = & C^{\mathcal{I}_i} \cup D^{\mathcal{I}_i} \\
(\neg_i C)^{\mathcal{I}_i} & = & \Delta^{\mathcal{I}_i} \backslash C^{\mathcal{I}_i} \\
(\neg_j C)^{\mathcal{I}_i} & = & r_{ji}((\neg_j C)^{\mathcal{I}_j}), \text{for } i \neq j \\
(\exists R.C)^{\mathcal{I}_i} & = & \{x \in \Delta^{\mathcal{I}_i} | \exists y, \langle x, y\rangle \in R^{\mathcal{I}_i} \wedge y \in C^{\mathcal{I}_i}\} \\
(\forall R.C)^{\mathcal{I}_i} & = & \{x \in \Delta^{\mathcal{I}_i} | \forall y, \langle x, y\rangle \in R^{\mathcal{I}_i} \rightarrow y \in C^{\mathcal{I}_i}\} \\
(\geqslant R.C)^{\mathcal{I}_i} & = & \{x \in \Delta^{\mathcal{I}_i} | \#\{y | \langle x, y\rangle \in R^{\mathcal{I}_i} \wedge y \in C^{\mathcal{I}_i}\} \geqslant n\} \\
(\leqslant R.C)^{\mathcal{I}_i} & = & \{x \in \Delta^{\mathcal{I}_i} | \#\{y | \langle x, y\rangle \in R^{\mathcal{I}_i} \wedge y \in C^{\mathcal{I}_i}\} \leqslant n\}
\end{array}
$$

Table 1: Local Interpretations

$\mathcal{ALCP}_\mathcal{C}$, a restricted $\mathcal{ALCP}$ that allows importing of only concept names.

Decidability requires that the reuse of role names be restricted such that a *locally simple role* that is used in number restriction will not be declared as a super-role of a transitive role in any ontology module. (A locally simple role is a role that is not transitive nor has any transitive sub-role in its home package). In practice, it is usually hard to check if imported role names and all their super-roles in the importing transitive closure are not used in number restrictions. However, decidability can be ensured by requiring a stronger condition that is easy to check: *A locally non-simple role should not be declared as a sub-role of an imported role or its inverse*.

2.2 Semantics

Definition 2 *A $\mathcal{SHOIQP}$ KB has* localized semantics *in that each package has its own local interpretation domain. Formally, for a $\mathcal{SHOIQP}$ KB $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j}\rangle$, a* distributed interpretation *is $\mathcal{I} = \langle \{\mathcal{I}_i\}, \{r_{ij}\}_{i \neq j}\rangle$, where $\mathcal{I}_i = \langle \Delta^{\mathcal{I}_i}, (.)^{\mathcal{I}_i}\rangle$ is the* local interpretation *of package P_i; $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ is the* domain relation *from P_i to P_j. For convenience, we may also denote $r_{ii} = \{(x, x) | \forall x \in \Delta^{\mathcal{I}_i}\}$ as the identity mapping in local domain $\Delta^{\mathcal{I}_i}$. For a subset S of $\Delta^{\mathcal{I}_i}$, $r_{ij}(S) = \{y | \exists x \in S, \langle x, y\rangle \in r_{ij}\}$.*

Each local interpretation $\mathcal{I}_i$ has a local domain $\Delta^{\mathcal{I}_i}$, and an interpretation function $(.)^{\mathcal{I}_i}$ which maps every concept name to a subset of $\Delta^{\mathcal{I}_i}$, every role name to a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$, and every individual name to an element in $\Delta^{\mathcal{I}_i}$, such that equations in Table 1 are satisfied.

A local interpretation $\mathcal{I}_i$ satisfies *a role inclusion axiom $R_1 \sqsubseteq R_2$ iff $R_1^{\mathcal{I}_i} \subseteq R_2^{\mathcal{I}_i}$, and it satisfies a GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$. An interpretation $\mathcal{I}_i$ is said to be a* model *of P_i (denoted by $\mathcal{I}_i \vDash P_i$), if it satisfies all of the axioms in P_i.*

The proposed semantics of $\mathcal{SHOIQP}$ is motivated by the need to overcome some of the limitations of existing approaches that can be traced to the arbitrary construction of domain relations and the lack of support for contextualized interpretation. Specifically, we seek a semantics that satisfies the following desiderata:

- **The preservation of concept unsatisfiability**. An unsatisfiable concept formula should not be reusable so as to be interpreted as a satisfiable concept. DDL, in its current form, does not preserve concept unsatisfiability due to the fact that a domain relation r_{ij} can map two disjoint non-empty subsets S_1, S_2 of $\Delta^{\mathcal{I}_i}$ to a non-empty set $r_{ij}(S_1) \cap r_{ij}(S_2)$. Formally, we say a domain relation r_{ij} *preserves the unsatisfiability* of an i-concept C if it is the case that whenever $C^{\mathcal{I}_i} = \emptyset$, it is necessarily the case that $C^{\mathcal{I}_j} = \emptyset$.
- **The transitive reusability of knowledge**. It should be possible to propagate the consequences of some of the axioms defined in one module in a transitive fashion to other ontology modules. For example, if a package P_i asserts that $C \sqsubseteq D$, and P_j (directly or indirectly) imports P_i, then it should be the case that $C \sqsubseteq D$ from the *point of view* of P_j.
- **Contextualized interpretation of knowledge**. The interpretation of assertions in each ontology module is constrained by its context. When knowledge in a module is reused by other modules, the interpretation of the reused knowledge should be constrained by the context in which the knowledge is reused.
- **Improved expressivity**. The language should support: (1) Inter-module concept inclusion and role inclusion (supported by DDL but not $\mathcal{E}$-Connections); (2) Concept construction using foreign concepts (supported by $\mathcal{E}$-Connection but not DDL); and (3) More general reuse of roles and of nominals than is supported by existing approaches.

We now proceed to explore the constraints that need to be placed on local interpretations to ensure that the resulting semantics for $\mathcal{SHOIQP}$ satisfies the desiderata enumerated above.

Definition 3 *An interpretation $\mathcal{I} = \langle \{\mathcal{I}_i\}, \{r_{ij}\}_{i \neq j}\rangle$ is a* model *of a $\mathcal{SHOIQP}$ KB $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j}\rangle$ if the following conditions are satisfied.*

1. *For any i, j, r_{ij} is one-to-one, i.e., it is an injective partial function.*
2. *For any i, j, k, r_{ij} is compositionally consistent, i.e., $r_{kj} \circ r_{ik} = r_{ij}$.*
3. *For every atomic i-concept C that appears in P_j ($i \neq j$), we have $r_{ij}(C^{\mathcal{I}_i}) = C^{\mathcal{I}_j}$.*
4. *For every atomic i-role p that appears in P_j ($i \neq j$), for every $(x, y) \in p^{\mathcal{I}_j}$, we have $(r_{ij}^-(x), r_{ij}^-(y)) \in p^{\mathcal{I}_i}$, and for every $z \in \Delta^{\mathcal{I}_i}$, we have:*
   - *forward closure: $(r_{ij}^-(x), z) \in p^{\mathcal{I}_i}$ iff $(x, r_{ij}(z)) \in p^{\mathcal{I}_j}$*
   - *backward closure: $(z, r_{ij}^-(y)) \in p^{\mathcal{I}_i}$ iff $(r_{ij}(z), y) \in p^{\mathcal{I}_j}$*
5. *For every i-nominal o that appears in P_j ($i \neq j$), $(o^{\mathcal{I}_i}, o^{\mathcal{I}_j}) \in r_{ij}$.*
6. *$\mathcal{I}_i \vDash P_i$ (for all i).*

The proposed semantics for $\mathcal{SHOIQP}$ is an extension of the semantics of $\mathcal{ALCP}_\mathcal{C}$ (Bao, Caragea, & Honavar 2006c) which relies on conditions 1, 2, 3, 6 above, and Semantic Importing (Pan, Serafini, & Zhao 2006) approach in which condition 4 was first introduced. In section 4, we will show that the conditions 1-6 are *necessary* and *sufficient* to ensure that the desiderata we outlined above for the semantics of $\mathcal{SHOIQP}$ are indeed satisfied.

In what follows, we will use $r_{ij}(f^{\mathcal{I}_i}) = f^{\mathcal{I}_j}$ to denote the relation between the local interpretations $f^{\mathcal{I}_i}$ and $f^{\mathcal{I}_j}$ of an atomic i-formula f.

Definition 4 *A closed knowledge base* Σ *is* consistent *as witnessed by a package* P_i *of* Σ *if* P_i^* *has a model. A concept* C *is* satisfiable *as witnessed by a package* P_i *if there is a model of the knowledge base* P_i^* *such that* $C^{\mathcal{I}_i} \neq \emptyset$. P_i *witnesses a concept subsumption* $C \sqsubseteq D$ *(denoted by* $C \sqsubseteq_i D$*), if for every model of* P_i^*, $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.

Hence, in $\mathcal{SHOIQP}$, consistency, satisfiability and subsumption problems are always answered *from the point of view of the witness package*, and it is possible that different packages draw different conclusions from their own points of view.

2.3 $\mathcal{SHOIQP}$ Examples

The semantic importing approach described here can model a broad range of scenarios that can be modeled using existing approaches.

Example 1: *Inter-module concept and role inclusions.* Suppose we have a *people* ontology P_1:

$$\begin{aligned} \top_1 &\sqsubseteq 1:\mathsf{Man} \sqcup 1:\mathsf{Woman} \\ 1:\mathsf{Boy} \sqcup 1:\mathsf{Girl} &\sqsubseteq 1:\mathsf{Child} \\ 1:\mathsf{Husband} &\sqsubseteq 1:\mathsf{Man} \sqcap \exists 1:\mathsf{marriedTo}.1:\mathsf{Woman} \end{aligned}$$

Suppose a *work* ontology P_2 imports some of the knowledge from the *people* ontology:

$$\begin{aligned} 1:\mathsf{marriedTo} &\sqsubseteq 2:\mathsf{knows} && (1) \\ 2:\mathsf{FemaleEmployee} &\sqsubseteq 2:\mathsf{Employee} && (2) \\ 2:\mathsf{MaleEmployee} &\sqsubseteq 2:\mathsf{Employee} && (3) \\ 2:\mathsf{MaleEmployee} &\sqsubseteq 1:\mathsf{Man} && (4) \\ 2:\mathsf{FemaleEmployee} &\sqsubseteq 1:\mathsf{Woman} && (5) \\ 1:\mathsf{Child} &\sqsubseteq \neg_2 2:\mathsf{Employee} && (6) \end{aligned}$$

Axiom (1) models inter-module role inclusion and (4-5) models inter-module concept inclusions. It also shows the semantic importing approach can realize concept specialization (4-5) and generalization (6).

Example 2: *Use of foreign roles or foreign concepts to construct local concepts.* Suppose a *marriage* ontology P_3 reuses the *people* ontology:

$$(=1\ (1:\mathsf{marriedTo}).(1:\mathsf{Woman})) \sqsubseteq 3:\mathsf{Monogamist} \quad (7)$$

$$3:\mathsf{MarriedPerson} \sqsubseteq \forall(1:\mathsf{marriedTo}).(3:\mathsf{MarriedPerson}) \quad (8)$$

$$3:\mathsf{NuclearFamily} \sqsubseteq \exists(\mathsf{hasMember}).(1:\mathsf{Child}) \quad (9)$$

A complex concept in P_3 may be constructed using an imported role (8), an imported concept (9), or both an imported role and an imported concept (7).

Example 3: *The use of nominals.* Suppose the *work* ontology P_2 defined above is augmented with additional knowledge from a *calendar* ontology P_4, to obtain an augmented *work* ontology. Suppose P_4 contains the following axiom:

$$4:\mathsf{WeekDay} = \{4:Mon, 4:Tue, 4:Wed, 4:Thu, 4:Fri\}$$

where the nominals are shown in *italic* font. Suppose the new version of P_2 contains the following additional axioms:

$$\begin{aligned} \{4:Fri\} &\sqsubseteq \exists 2:\mathsf{hasDressingCode}.2:\mathsf{CasualDress} \\ \top_2 &\sqsubseteq 2:\mathsf{hasDressingCode}^-.(4:\mathsf{WeekDay}) \end{aligned}$$

3 Reduction to Ordinary DL

A reduction $\Re$ from a closed $\mathcal{SHOIQP}$ KB $\Sigma_d = \langle\{P_i\}, \{P_i \mapsto P_j\}_{i\neq j}\rangle$ to a $\mathcal{SHOIQ}$ KB Σ' can be obtained as follows:

- The signature of Σ' is the union of the local signatures of the component packages, i.e. $\bigcup_i \mathsf{Loc}(P_i)$
- Σ' is constructed such that: $\forall i$, the concepts $\top_i \in \Sigma'$; $\top, \bot \in \Sigma'$.
- $\forall i, j, k$ such that $P_i \in I(P_j)$, $P_i \in I(P_k)$ and $P_k \in I(P_j)$, add $\top_i \sqcap \top_j \sqsubseteq \top_k$ to Σ'.
- Copy each GCI or role inclusion $X \sqsubseteq Y$ in P_i as $\#(X) \sqsubseteq \#(Y)$. The mapping #() is defined below.
- For each local atomic concept or nominal C in P_i, add $i:C \sqsubseteq \top_i$ to Σ'.
- For each local atomic role P in P_i, add $\top_i$ as its domain and range, i.e. add $\top \sqsubseteq \forall P^-.\top_i$ and $\top \sqsubseteq \forall P.\top_i$ to Σ'.
- For each imported atomic role P in P_i, add the following axioms to Σ':
  - $\exists P.\top_i \sqsubseteq \forall P.\top_i$ (forward closure)
  - $\exists P^-.\top_i \sqsubseteq \forall P^-.\top_i$ (backward closure)
  - $\#(P) \sqsubseteq P$
  - $\exists\#(P).\top_i \sqsubseteq \top_i$ (local domain)
  - $\exists\#(P)^-.\top_i \sqsubseteq \top_i$ (local range)
  - $Trans(\#(P))$ if P is transitive

The mapping #() is adapted from a similar one for DDL (Borgida & Serafini 2003) with the modifications needed to allow name importing. For a formula X used in P_j, $\#(X)$ is:

- X, for an atomic j-term X.
- $X \sqcap \top_j$, for an atomic pure i-concept or i-nominal X ($i \neq j$).
- $\neg\#(X) \sqcap \top_j$, for $\neg_i X$ where X is a pure i-concept ($i \neq j$).
- $X^{i\to j}$, for an atomic i-role X ($i \neq j$), where $X^{i\to j}$ is a new "image" role name.
- $\top_j \sqcap \top_i \sqcap \rho(\#(X_1), ..., \#(X_k))$, for an i-concept $X = \rho(X_1, ..., X_k)$, where ρ is a concept constructor with k arguments.

For example:

$$\begin{aligned} \#(j:(\neg_i i:C)) &= \top_j \sqcap \top_i \sqcap \neg C \\ \#(j:(j:D \sqcup i:C)) &= \top_j \sqcap ((\top_j \sqcap D) \sqcup (\top_j \sqcap C)) \\ \#j:(\forall(j:P).(i:C)) &= \top_j \sqcap \forall P.(\top_j \sqcap C) \\ \#j:(\exists(i:P).(i:C)) &= \top_j \sqcap \top_i \sqcap \exists P^{i\to j}.(\top_j \sqcap C) \end{aligned}$$

It should be noted that #() is *contextualized* so as to allow a formula with the same syntax to have different translations when it appears in different packages.

4 Properties of Semantic Importing

In this section, we further justify the proposed semantics for $\mathcal{SHOIQP}$. Specifically, we summarize our main results which show that $\mathcal{SHOIQP}$ satisfies the desiderata summarized in section 2. (Complete proofs are given in a technical report (Bao, Slutzki, & Honavar 2007)).

Formally, we have:

Lemma 1 *A $\mathcal{SHOIQP}$ KB Σ is consistent as witnessed by a package P_i iff $\Re(P_i^*)$ is consistent.*

Proof sketch: For every (distributed) model of Σ, we can always construct a (classical) model for $\Re(P_i^*)$, and vice versa.

Theorem 1 (Reasoning Exactness) *For a $\mathcal{SHOIQP}$ KB $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j} \rangle$, $C \sqsubseteq_i D$ iff $\Re(P_i^*) \models \#(C) \sqsubseteq \#(D)$.*

Corollary 1 (The Preservation of Unsatisfiability) *For a $\mathcal{SHOIQP}$ KB $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j} \rangle$, $P_i \in I(P_j)$, if $C \sqsubseteq_i \bot$ then $C \sqsubseteq_j \bot$.*

Theorem 2 (Monotonicity) *For a $\mathcal{SHOIQP}$ KB $\Sigma = \langle \{P_i\}, \{P_i \mapsto P_j\}_{i \neq j} \rangle$, if $P_i \in I(P_j)$ and $C \sqsubseteq_i D$, then $C \sqsubseteq_j D$, where $\mathsf{Sig}(C)$ and $\mathsf{Sig}(D)$ are subsets of $\mathsf{Sig}(P_i) \cap \mathsf{Sig}(P_j)$.*

Theorem 2 ensures that when some part of an ontology module is reused, the restrictions asserted by it (e.g. domain restrictions of roles) will not be relaxed to prohibit the reuse of imported knowledge. Theorem 2 also ensures that consequences of imported knowledge can be transitively propagated to all packages that are reachable from the source of the imported knowledge via a chain of importing relations.

From the proofs of Lemma 1 and Theorem 2, we have:

Lemma 2 *For every concept C such that $\mathsf{Sig}(C) \subseteq \mathsf{Sig}(P_i) \cap \mathsf{Sig}(P_j)$ where P_i, P_j are two packages and $P_i \in I(P_j)$, we have $r_{ij}(C^{\mathcal{I}_i}) = C^{\mathcal{I}_j}$.*

Finally, the semantics of $\mathcal{SHOIQP}$ ensures that the interpretation of axioms in an ontology module is constrained by their *contexts*, as seen from the reduction to a corresponding integrated ontology: $C \sqsubseteq D$ in P_i is mapped to $\top_i \sqcap \#(C) \sqsubseteq \top_i \sqcap \#(D)$.

When an i-GCI is propagated to module P_j, it will only affect the "shared" domain $r_{ij}(\Delta^{\mathcal{I}_i})$, and not the entire domain $\Delta^{\mathcal{I}_j}$. Suppose package P_i contains an axiom $\neg_i \mathsf{Male} \sqsubseteq \mathsf{Female}$. i.e., every individual is either male or female. Suppose package P_j imports P_i. Then in P_j, it need not be the case that $\top_j \sqsubseteq \mathsf{Male} \sqcup \mathsf{Female}$. This is because $r_{ij}(\Delta^{\mathcal{I}_i}) \subseteq \Delta^{\mathcal{I}_j}$, $\Delta^{\mathcal{I}_i} \backslash \mathsf{Male}^{\mathcal{I}_i} \subseteq \mathsf{Female}^{\mathcal{I}_i}$. That is, $\Delta^{\mathcal{I}_i} = \mathsf{Male}^{\mathcal{I}_i} \cup \mathsf{Female}^{\mathcal{I}_i}$ does not necessarily mean $\Delta^{\mathcal{I}_j} = \mathsf{Male}^{\mathcal{I}_j} \cup \mathsf{Female}^{\mathcal{I}_j}$. This example illustrates the importance of contextualizing negation as well as 'top' to preserve the original meaning of imported knowledge.

Hence, the effect of an axiom is always contextualized within its original designated context. Therefore, it is not necessary to explicitly restrict the use of ontology language to ensure locality of axioms as required by conservative extension (Grau *et al.* 2007). The locality of axioms follows from the semantics of $\mathcal{SHOIQP}$.

The constraints on domain relations given in Definition 3 on the semantics of $\mathcal{SHOIQP}$ are minimal in the sense that if we drop any of the six requirements, we can no longer satisfy some of the desiderata summarized in section 2.2. In the absence of requirements 3, the reuse of concept names will be just syntactical. Thus, the local interpretations of shared concept names can be determined independently and hence may be inconsistent with each other. Requirement 5 is needed to ensure that each nominal has a unique instance (which may be "copied" by multiple local interpretations associated by domain relations). Requirement 6 is natural because constraints within each module must be satisfied.

Dropping requirement 1 (one-to-one domain relation) leads to difficulties in preservation of concept unsatisfiability. For example, if domain relations are not injective, then $C_1 \sqsubseteq_i \neg_i C_2$, $D \sqsubseteq_j C_1$ and $D \sqsubseteq_j C_2$ does not ensures $D \sqsubseteq_j \bot$. If domain relations are not partial functions, an individual in $\Delta^{\mathcal{I}_i}$ may get mapped to different individuals in $\Delta^{\mathcal{I}_j}$ via r_{ij}. In this case, preservation of unsatisfiability of a complex concept can no longer be guaranteed when both number restriction and role importing are allowed.

Dropping requirement 2 (compositional consistency of domain relations) would result in violation of monotonicity of inference based on imported knowledge in general, and transitive reusability requirement in particular. In the absence of compositional consistency of domain relations, the importing relations will be like bridge rules in DDL in that they are localized w.r.t. the connected pairs of modules without support for propagation of knowledge across a succession of modules connected by importing relations.

If requirement 4 (forward and backward closure of role instances) is dropped, we can no longer ensure the consistency of local interpretations of complex concepts constructed from number restrictions. It ensures the numbers of p-successors and p-predecessors of an individual are always kept the same as that in the interpretation of role p's home package.

5 Summary and Discussion

In this paper, we introduced a modular ontology language $\mathcal{SHOIQP}$ to reuse knowledge from multiple ontology modules. A $\mathcal{SHOIQP}$ ontology consists of multiple ontology modules (each of which can be viewed as a $\mathcal{SHOIQ}$ ontology) and concept, role and nominal names can be shared by "importing" relations among modules.

The proposed language supports contextualized interpretation, i.e., interpretation from the point of view of a specific package. We have established the necessary and sufficient constraints on domain relations (i.e., the relations between individuals in different local domains) to preserve concept unsatisfiability, monotonicity of inference, and transitive reuse of knowledge.

We have shown in the extended version of the paper (Bao, Slutzki, & Honavar 2007) that several other modular ontology formalisms can be simulated by $\mathcal{SHOIQP}$ including in particular, *DDL* with homogeneous bridge rules between concepts and between roles, and *one-way $\mathcal{E}$-Connections* between $C^{\mathcal{E}}_{\mathcal{IHQ}}(\mathcal{SHOIN})$ ontologies (Kutz *et al.* 2004; Grau, Parsia, & Sirin 2004). However, because DDL does not allow us to impose constraints on domain relations (e.g., compositional consistency which is necessary in $\mathcal{SHOIQP}$), $\mathcal{SHOIQP}$ cannot be reduced to DDL. It should also be noted that DDL with heterogeneous bridge rules cannot be reduced to $\mathcal{SHOIQP}$. $\mathcal{SHOIQP}$ also provides some modeling ability not offered by $\mathcal{E}$-Connections in its current form, e.g., the use of foreign roles to define local concepts, and the definition of role inclusion between a foreign roles and a local role.

Distributed representation and reasoning in multiple knowledge bases is addressed in (Kaneiwa & Mizoguchi 2004) using an order-sorted logic. In contrast, our focus is on modular description logics. Recent work on semantics of ontology versioning (Heflin & Pan 2004) has explored the problem of supporting reasoning based on different *versions* of an ontology. In that setting, a newer version of an ontology can be viewed as importing knowledge from an older version. On the other hand, the main concern of our work is knowledge reuse, which is more general than ontology versioning.

The proposed $\mathcal{SHOIQP}$ improves the P-DL $\mathcal{ALCP}_{\mathcal{C}}$ by (Bao, Caragea, & Honavar 2006c) and a related proposal for semantic importing introduced by (Pan, Serafini, & Zhao 2006) in several significant aspects:

- **Increased expressivity**: provided by support for the use of $\mathcal{SHOIQ}$ instead of the much more restricted $\mathcal{ALC}$ by individual modules and by support for concept, role and nominal importing (unlike in the case of P-DL $\mathcal{ALCP}_{\mathcal{C}}$ which only allows concept importing).
- **Contextualized negation**: a necessary condition for preservation of unsatisfiability
- **Monotonicity**: a property not guaranteed by the semantic importing approach of (Pan, Serafini, & Zhao 2006).

Ongoing work is aimed at developing a distributed reasoning algorithm for $\mathcal{SHOIQP}$ by extending the results of (Bao, Caragea, & Honavar 2006c) and (Pan, Serafini, & Zhao 2006).

Acknowledgement: This research was supported in part a grant from the US NSF (IIS-0639230).

References

Bao, J.; Caragea, D.; and Honavar, V. 2006a. Modular ontologies - a formal investigation of semantics and expressivity. In *Asian Semantic Web Conference (ASWC)*, 616–631.

Bao, J.; Caragea, D.; and Honavar, V. 2006b. On the semantics of linking and importing in modular ontologies. In *International Semantic Web Conference (ISWC)*, 72–86.

Bao, J.; Caragea, D.; and Honavar, V. 2006c. A tableau-based federated reasoning algorithm for modular ontologies. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 404–410. IEEE Press.

Bao, J.; Slutzki, G.; and Honavar, V. 2007. A semantic importing approach to knowledge reuse from multiple ontologies (extended version). Technical report, TR-539 Computer Science, Iowa State University.

Borgida, A., and Serafini, L. 2003. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics* 1:153–184.

Bouquet, P.; Giunchiglia, F.; van Harmelen, F.; Serafini, L.; and Stuckenschmidt, H. 2003. C-OWL: Contextualizing ontologies. In Fensel, D.; Sycara, K. P.; and Mylopoulos, J., eds., *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, 164–179. Springer.

Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics. In *KR*, 187–197.

Grau, B. C., and Kutz, O. 2007. Modular ontology languages revisited. In *Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa), co-located with IJCAI*.

Grau, B. C.; Horrocks, I.; Kutz, O.; and Sattler, U. 2006. Will my ontologies fit together? In *Proc. of the 2006 Description Logic Workshop (DL 2006)*, volume 189.).

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007. A logical framework for modularity of ontologies. In *IJCAI*, 298–303.

Grau, B. C.; Parsia, B.; and Sirin, E. 2004. Working with multiple ontologies on the semantic web. In *International Semantic Web Conference*, 620–634.

Heflin, J., and Pan, Z. 2004. A model theoretic semantics for ontology versioning. In *International Semantic Web Conference*, 62–76.

Horrocks, I., and Sattler, U. 2005. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. In *IJCAI*, 448–453.

Kaneiwa, K., and Mizoguchi, R. 2004. Ontological knowledge base reasoning with sort-hierarchy and rigidity. In *KR*, 278–288.

Kutz, O.; Lutz, C.; Wolter, F.; and Zakharyaschev, M. 2004. E-connections of abstract description systems. *Artif. Intell.* 156(1):1–73.

Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative extensions in expressive description logics. In *IJCAI*, 453–458.

Pan, J.; Serafini, L.; and Zhao, Y. 2006. Semantic import: An approach for partial ontology reuse. In *1st International Workshop on Modular Ontologies (WoMo 2006), co-located with ISWC*.