2006

On the Semantics of Linking and Importing in Modular Ontologies

Jie Bao
Iowa State University

Doina Caragea
Iowa State University

Vasant Honavar
Iowa State University

Recommended Citation

Bao, Jie; Caragea, Doina; and Honavar, Vasant, "On the Semantics of Linking and Importing in Modular Ontologies" (2006). *Computer Science Technical Reports*. 246.
http://lib.dr.iastate.edu/cs_techreports/246

On the Semantics of Linking and Importing in Modular Ontologies

Abstract

Modular ontology languages, such as Distributed Description Logics (DDL), E-connections and Package-based Description Logics (P-DL) offer two broad classes of approaches to connect multiple ontology modules: the use of mappings or linkings between ontology modules e.g., DDL and E-connections; and the use of importing e.g., P-DL. The major difference between the two approaches is on the usage of "foreign terms" at the syntactic level, and on the local model disjointness at the semantic level. We compare the semantics of linking in DDL and E-connections, and importing in P-DL within the Distributed First Order Logics (DFOL) framework. Our investigation shows that the domain disjointness assumption adopted by the linking approach leads to several semantic difficulties. We explore the possibility of avoiding some of these difficulties using the importing approach to linking ontology modules.

Disciplines

Artificial Intelligence and Robotics

On the Semantics of Linking and Importing in Modular Ontologies [⋆]

Jie Bao[1], Doina Caragea[2], Vasant G Honavar[1]

[1]Artificial Intelligence Research Laboratory,
Department of Computer Science,
Iowa State University,
Ames, IA 50011-1040, USA
[2]Department of Computing and Information Sciences
Kansas State University,
Manhattan, KS 66506, USA
[1]{baojie, honavar}@cs.iastate.edu, [2]dcaragea@ksu.edu

Abstract. Modular ontology languages, such as Distributed Description Logics (DDL), $\mathcal{E}$-connections and Package-based Description Logics (P-DL) offer two broad classes of approaches to connect multiple ontology modules: the use of *mappings* or *linkings* between ontology modules e.g., DDL and $\mathcal{E}$-connections; and the use of *importing* e.g., P-DL. The major difference between the two approaches is on the usage of "foreign terms" at the syntactic level, and on the local model disjointness at the semantic level. We compare the semantics of linking in DDL and $\mathcal{E}$-connections, and importing in P-DL within the Distributed First Order Logics (DFOL) framework. Our investigation shows that the domain disjointness assumption adopted by the linking approach leads to several semantic difficulties. We explore the possibility of avoiding some of these difficulties using the importing approach to linking ontology modules.

1 Introduction

Because the web is a network of loosely coupled, distributed, autonomous entities, it is inevitable that the ontologies on the web to be modular, collaboratively built and partially connected. However, the current web ontology language OWL provides only limited support for modular ontologies. Consequently, it fails to accommodate localized semantics, partial reuse, selective knowledge hiding, and scalable inference [3]. Hence, there is significant interest on modular ontology languages, such as, Distributed Description Logics (DDL) [4], $\mathcal{E}$-connections [13,10] and Package-based Description Logics (P-DL) [3].

These proposals adopt two broad classes of approaches to asserting semantic relations between multiple ontology modules: the use of *mappings* or *linkings* between ontology modules e.g., DDL and $\mathcal{E}$-connections; and the use of *importing* e.g., P-DL. The major difference between the two approaches has to do with

[⋆] A shorter version of the report is published in the 5th International Semantic Web Conference (ISWC 2006) the use of "foreign terms" in ontology modules. In a linked ontology, different modules have *disjoint terminologies* and *disjoint interpretation* domains, and semantic relations between ontology modules are only enabled by a set of *mapping axioms*, such as bridge rules in DDL or $\mathcal{E}$-connections. Therefore, the direct usage of terms defined in one module is forbidden in another module. In contrast, *importing* allows an ontology module to make direct reference to terms defined in other ontology modules, i.e., *importing of foreign terms*.

Serafini *et.al.* (2005) [18] compare mapping or linking based approaches to the "integration" of multiple ontology modules such as DDL and $\mathcal{E}$-connections by reducing them to the Distributed First Order Logics (DFOL) [7] framework. However, there is little work on the formal investigation of the importing approach to integrating ontology modules. Against this background, we compare the semantics of the two approaches within the DFOL framework, with the study of their strengthes and limitations. Such an investigation reveals that the importing approach, with the removing of the module disjointness assumption adopted by the linking approach, can provide stronger expressivity and avoid many of the semantic difficulties in current modular ontology language proposals.

The rest of the paper is organized as follows. Section 2 introduces the requirements of modular ontology languages. Section 3 introduces DFOL. Section 4 (DDL) and 5 ($\mathcal{E}$-Connections) describe the semantics and limitations of linking approaches. Section 6 shows the semantics of importing in OWL and why it is only a syntactic, not semantic solution for modularity. Section 7 presents an improved importing approach used in P-DL. Section 8 concludes with a summary of the paper and a discussion of some additional issues related to modular ontologies.

2 Desiderata For Modular Ontology Languages

We first list a set of minimal requirements for modular ontology languages [2] on the semantic web as the basis for our comparison of the semantics of DDL, $\mathcal{E}$-connections and P-DL within the DFOL framework:

1. **Localized Semantics**. A modular ontology should not only be *syntactically modular* (e.g. stored in separated XML name spaces), but also *semantically modular*. That is, the existence of a *global model* should not be a requirement for integration of ontology modules.
2. **Exact Reasoning**. The answer to a reasoning problem over a collection of ontology modules should be *semantically equivalent* to that obtained by reasoning over an ontology resulting from an *integration* of the relevant ontology modules. Thus, if an ontology O contains $A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D$, and a modularized version of O has two modules $M_1 = \{A \sqsubseteq B\}$, $M_2 = \{C \sqsubseteq D\}$ and a semantic connection $r(B, C)$, which represents the modularized version of $B \sqsubseteq C$, the answer to any reasoning problem obtained by integration of M_1, M_2 and $r(B, C)$ should be the same as that obtained by using a sound and complete reasoner on O.

3. **Directed Semantic Relations**. The framework must support *directional semantic relations* from a *source* module to a *target* module. A directional semantic relation affects only the reasoning within the target module and not the source module.
4. **Transitive Reusability**. Knowledge contained in ontology modules should be directly or indirectly reusable. That is, if a module A reuses module B, and module B reuses module C, then effectively, module A reuses module C.

Other desiderata that have been considered in the literature include: the ability to cope with local inconsistency or global inconsistency, and local logic completeness. We believe that the desiderata listed above are among the most critical ones for a modular ontology to be semantically sound and practically usable.

3 Distributed First Order Logics

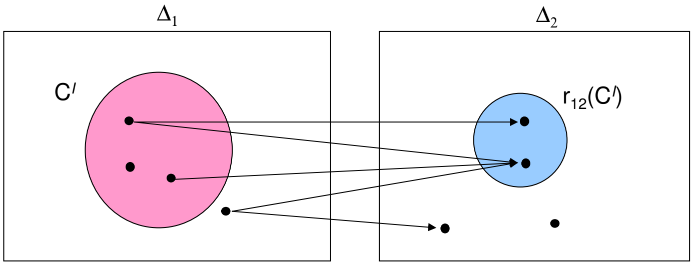
This section briefly reviews the Distributed First Order Logics (DFOL) framework introduced by Ghidini and Serafini. A DFOL knowledge base (KB) [7] (and hence, a DFOL ontology) includes a family of first order languages $\{L_i\}_{i\in I}$, defined over a finite set of indices I. We will use L_i to refer to the ith module of the ontology . An (i-)variable x or (i-)formula ϕ occurring in module L_i is denoted as $i : x$ or $i : \phi$ (we drop the prefix when there is no confusion). The signature (the set of all names) of L_i are i-terms.

The semantics of DFOL, in the light of Local Model Semantics [6], includes a set of local models and domain relations. For each L_i, there is an interpretation domain $\varDelta_i$. Let M_i be the set of all first order models of L_i on $\varDelta_i$. We call each $m \in M_i$ a *local model* of L_i. A *domain relation* r_{ij}, where $i \neq j$, is a subset of $\varDelta_i \times \varDelta_j$. The domain relation r_{ij} represents the capability of the module j to map the objects of $\varDelta_i$ in $\varDelta_j$, or, the j's *subjective view* of the relation between $\varDelta_i$ and $\varDelta_j$. In general, $r_{ij} \neq r_{ji}^-$.

We use $\langle d, d'\rangle$ in r_{ij} to denote that from the point of view of j, the object d in $\varDelta_i$ is *mapped* to the object d' in $\varDelta_j$; d is called a *pre-image* of d', and d' is called an *image* of d. In general, domain relations can be injective, surjective, bijective, or arbitrary (as shown in Figure 1). For an object $d \in \varDelta_i$, $r_{ij}(d)$ denotes the set $\{d' \in \varDelta_j | \langle d, d'\rangle \in r_{ij}\}$. For a subset $D \subseteq \varDelta_i$, $r_{ij}(D)$ denotes $\cup_{d\in D} r_{ij}(d)$, is the image set of D.

Example 1 *An ontology contains two modules $L_{\{1,2\}}$. L_1 contains knowledge about regions and their relations, such as $\forall x, Country(x) \rightarrow Region(x)$ (an 1-formula) (every country is a region). L_2 contains knowledge about people, such as $\exists x$, $European(x) \wedge Caucasian(x)$ (some European people are Caucasian). The local domain $\varDelta_1$ has objects India and USA, and local domain $\varDelta_2$ has objects Hindu, Indian, Iowan and American. The domain relation r_{12} is $\langle 1 : \mathsf{India}, 2 : \mathsf{Indian}\rangle, \langle 1 : \mathsf{USA}, 2 : \mathsf{American}\rangle, \langle 1 : \mathsf{USA}, 2 : \mathsf{Iowan}\rangle$, while the domain relation*

r_{21} *is* $\langle 2:\textsf{Hindu}, 1:\textsf{India}\rangle, \langle 2:\textsf{Indian}, 1:\textsf{USA}\rangle, \langle 2:\textsf{American}, 1:\textsf{USA}\rangle$. $r_{12}(1:\textsf{USA}) = \{2:\textsf{American}, 2:\textsf{Iowan}\}$.



A DFOL KB has two modules L_1 and L_2. The domain of a model of L_1 is Δ_1, and the domain of a model of L_2 is Δ_2. C is a unary predicate (i.e. concept in description logics) in L_1. The interpretation set $C^{\mathcal{I}}$ is the set of objects in Δ_i assigned to C. r_{12} is the domain relation. $r_{12}(C^{\mathcal{I}})$ is the image set of $C^{\mathcal{I}}$ in Δ_2. Note that an object in Δ_1 may have multiple images or no image in Δ_2 and an object in Δ_2 may have multiple pre-images or no pre-image in Δ_1.

Fig. 1. Semantics of DFOL

Note that L_1 and L_2 hold different semantic points of view, e.g. on the meaning of Indian, so that $r_{12} \neq r_{21}^{-}$. The example reveals that r_{ij} is j' subjective point of view, not an objective point of view from an external observer, and may not be symmetric to r_{ji}.

Some useful lemmas about domain relations are given here:

Lemma 1 *For a domain relation r_{ij}, $i \neq j$, for any unary predicate C, D of i, if $\forall x, C(x) \rightarrow D(x)$ (in DL, $C \sqsubseteq D$), then $r_{ij}(C^{m_i}) \subseteq r_{ij}(D^{m_i})$*

Lemma 2 *For a domain relation r_{ij}, $i \neq j$, for any set C^{m_i}, D^{m_i}, $r_{ij}(C^{m_i} \cap D^{m_i}) \subseteq r_{ij}(C^{m_i}) \cap r_{ij}(D^{m_i})$.*

Lemma 3 *For a domain relation r_{ij}, $i \neq j$, for any set C^{m_j}, D^{m_j}, $r_{ij}^{-}(C^{m_j} \cap D^{m_j}) \subseteq r_{ij}^{-}(C^{m_j}) \cap r_{ij}^{-}(D^{m_j})$.*

In more general sense, an ontology specified with DFOL is a set of individual descriptions for the same domain of discourse, from a set of correlated but not identical points of view of multiple observers, or agents. Each ontology module is the mind state held by an agent of its reading of the identification of objects in the domain and relations among these objects. A domain relation r_{ij} is the ability of the agent j to explain the mind state of agent i. For semantically sound reasoning to be possible from any module (agent)'s point of view, domain relations should be consistent and do not lead to semantic ambiguity. We will show in the following text how arbitrary domain relations can lead to semantic imprecision and reasoning difficulties and principled approaches to avoid such problems.

4 Semantics of Linking in DDL

One influential family of modular ontology formalisms is the linking approach. The linking approach is aimed at preserving the autonomy of loosely coupled modules, while allowing restricted "mappings" between formulae of linked modules. Formally, a linking approach holds the follow assumptions:

- For any L_i and $L_j, i \neq j$, i-terms and j-terms are disjoint.
- The semantic connection between L_i and L_j is enabled only by mapping rules between i-terms and j-terms, which are interpreted as domain relations $r_{ij} \subseteq \Delta_i \times \Delta_j$.
- Local interpretation domains and domain relations are disjoint. For any $i \neq j$, $\Delta_i \times \Delta_i$ (or $\Delta_j \times \Delta_j$) has intersection neither with r_{ij} nor with r_{ji}.

Based on DFOL, Distributed Description Logics (DDL) [4] with its syntax C-OWL [5] (with some extended features to the original DDL proposal), is one of the first linking-based modular ontology formalisms. In DDL, the semantic mappings between disjoint modules L_i and L_j are established by a set of inter-module axioms called "Bridge Rules"(B_{ij}) of the form:

- INTO rule: $i : \phi \xrightarrow{\sqsubseteq} j : \psi$, semantics: $r_{ij}(\phi^{m_i}) \subseteq \psi^{m_j}$
- ONTO rule: $i : \phi \xrightarrow{\sqsupseteq} j : \psi$, semantics: $r_{ij}(\phi^{m_i}) \supseteq \psi^{m_j}$

where $m_i(m_j)$ is a model of $L_i(L_j)$, ϕ, ψ are formulae, r_{ij} is a domain relation which serves as the interpretation of B_{ij}. Note that B_{ij} is directional. We will only consider bridge rules between concepts, not roles [5], since there is still no reasoning support for role bridge rules [17].

Distributed concept correspondence between two modules in DDL covers some important scenarios that require mapping between ontology modules. However, the expressivity of DDL is limited in some settings that arise in practical applications: For example, DDL cannot be used to express "a person x works in a region y". In general, it can not construct new concepts using terms across modules, such as restrictions $\forall 1 : R.2 : D$ and $\exists 1 : R.2 : D$, where C, D are concepts and R is role.

In addition to the expressivity limitations, DDL may present semantic difficulties in some situations. While DDL bridge rules are intended to simulate *concept inclusions* [4,5], arbitrary modelling with bridge rules may lead to undesired semantics, such as in the *Subsumption Propagation problem* and *Inter-module Unsatisfiability problem*, as noted in [10,8]:

Example 2 (Subsumption Propagation) *A KB Σ_d includes modules $L_{\{1,2,3\}}$, each with an empty TBox; bridge rules $B_{12} = \{1 : Bird \xrightarrow{\sqsupseteq} 2 : Fowl\}$, $B_{23} = \{2 : Fowl \xrightarrow{\sqsupseteq} 3 : Chicken\}$. The entailment problem $1 : Bird \xrightarrow{\sqsupseteq} 3 : Chicken$ cannot be answered since bridge rules B_{13} are not given, nor can be inferred.*

Note that bridge rules may be inferred between the *same* pair of modules. For example, if $1 : A \xrightarrow{\sqsubseteq} 2 : B$ and $2 : B \sqsubseteq 2 : C$, it can be inferred that $1 : A \xrightarrow{\sqsubseteq} 2 : C$. Intra-module subsumption may also be reused in some particular cases. For example, if $1 : A \sqsubseteq 1 : B$, $1 : A \xrightarrow{\sqsupseteq} 2 : C$ and $1 : B \xrightarrow{\sqsubseteq} 2 : D$, it can be inferred that $2 : C \sqsubseteq 2 : D$ [19]. However, Example 2 shows that in general bridge rules in DDLs are not transitively reusable, thereby are restricted for many application scenarios.

Example 3 (Inter-module Unsatisfiability[10,8]) *DDLs may not detect unsatisfiability across ontology modules. A KB Σ_d includes modules $L_{\{1,2\}}$, $L_1 = \{1 : Bird \sqsubseteq 1 : Fly\}$, $L_2 = \{2 : Penguin \sqsubseteq \top\}$, $B_{12} = \{1 : Bird \xrightarrow{\sqsupseteq} 2 : Penguin, 1 : \neg Fly \xrightarrow{\sqsupseteq} 2 : Penguin\}$. Penguin is still satisfiable in Σ_d since such a distributed model exists:*

- *m_1: $\Delta_1 = \{a, b\}$, $Bird^{m_1} = Fly^{m_1} = \{a\}, (\neg Fly)^{m_1} = \{b\}$*
- *m_2: $\Delta_2 = \{x\}$, $Penguin^{m_2} = \{x\}$*
- *r_{12}: $\{\langle a, x\rangle, \langle b, x\rangle\}$*

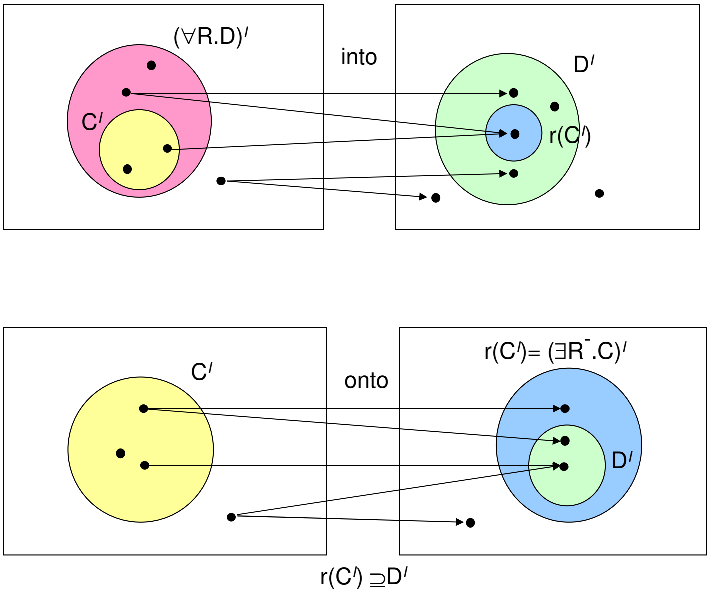
Such difficulties are rooted in the implicit local domain disjointness assumption of DDL: individuals in each local domain are private to that domain, and DDL semantics does not take into account if individuals in different local domains may represent the same physical world object. Therefore, a bridge rule, while intended to simulate concept inclusion, cannot be read directly as concept inclusion, such as $i : A \sqsubseteq j : B$. Instead, it must be read as a classic DL axiom in the following way [4]:

- $i : A \xrightarrow{\sqsubseteq} j : B \Rightarrow (i : A) \sqsubseteq \forall R_{ij}.(j : B)$
- $i : A \xrightarrow{\sqsupseteq} j : B \Rightarrow (j : B) \sqsubseteq \exists R_{ij}^{-}.(i : A)$

where R_{ij} is a new role representing correspondences B_{ij} between L_i and L_j. Such translations are best understood as shown in Figure 2.

Therefore, for the given subsumption propagation example, if $B_{13} = \emptyset$, entailment $Chicken \sqsubseteq \exists R_{13}^{-}.Bird$ is not always true. For the inter-module unsatisfiability problem, concept $Penguin$ ($\sqsubseteq \exists R_{12}^{-}.(Fly) \sqcap \exists R_{12}^{-}.(\neg Fly)$) is satisfiable.

Thus, the semantics of DDL are designed to simulate concept inclusion with a special type of roles, i.e., bridge rules. However, in the absence of a principled approach to avoid arbitrary domain relation interpretations for bridge rules, all semantic relations (bridge rules) between DDL modules are *localized* to pairs of modules that are bridged by the rules in question. Consequently, semantic relations between a pair of DDL modules cannot be safely reused by other modules, thereby precluding general subsumption propagation, and more generally, module transitive reusability. Note further that in order to enable distributed (not necessarily exact) reasoning in general, a DDL KB needs explicit declaration of domain relations between *each* pair of modules, leading to an exponential blowup in the number of bridge rules, with the attendant inefficiency and increased risk of inconsistencies.



(b) Onto Bridge Rules

Fig. 2. Semantics of DDL Bridge Rules

Serafini et al. [17] has asserted that the inter-module unsatisfiability difficulty is the result of incomplete modelling. They have argued that it can be eliminated if extra information, for example, $1 : \neg Bird \xrightarrow{\sqsubseteq} 2 : \neg Penguin$ and $1 : Fly \xrightarrow{\sqsubseteq} \neg 2 : Penguin$, is added to guarantee one-to-one domain relations. Our investigation reveals a more general result: one-to-one domain relations can guarantee that reasoning over DDL always yields the same result as that obtained from an integrated ontology when bridge rules are replaced with general concept inclusions (GCI). First, we have the definition:

Definition 1 *A domain relation r_{ij} for bridge rules B_{ij} is said to be* one-to-one *if for any bridge rule $C \xrightarrow{\sqsubseteq} D$ or $C \xrightarrow{\sqsupseteq} D$, for any $x \in C^{\mathcal{I}_i}$, there is one and only one unique $y \in \Delta_j$ such that $\langle x, y \rangle \in r_{ij}$.*

The integration process from a DDL ontology to a ordinary (global) DL ontology is given in [4]. For a DDL ontology $\{L_i\}$, the global DL (GDL) ontology is defined as follows:

- There is a new top concept $\top_g$ and a new bottom concept $\bot_g$ in GDL.
- The primitive concepts of GDL consist of $i : A$ obtained from primitive concepts or constant concepts A (such as $\top_i$ and $\bot_i$) of L_i
- The primitive roles of GDL include $i : p$ obtained from primitive or constant roles p of L_i

The mapping $\#()$ from concepts/roles in L_i to concepts/roles in GDL is defined as follows: for atomic concepts, roles, and individuals $i : M$, $\#(i : M) = i : M$; for a complex concept constructor ρ with k arguments, $\#(i : \rho(X_1, ..., X_k)) = \top_i \sqcap \rho(\#(X_1), ..., \#(X_k))$. For example, $\#i : (\forall p.C) = \top_i \sqcap \forall(i : p).(\top_i \sqcap i : C)$.

Applying $\#()$ to a DDL knowledge base $\Sigma = \langle\{L_i\}, \{B_{ij}\}\rangle$, we get an integrated GDL [4] $\#(\Sigma)$ that contains:

- $\#(i : A) \sqsubseteq \#(i : B)$ for all $i : A \sqsubseteq B \in L_i$
- $\bot_i \sqsubseteq \bot_g$
- $\#(i : A) \sqsubseteq \top_i$ for each atomic concept A of L_i
- Axioms that ensure the domain and range of any i-role to be $\top_i$: $\top_i \sqsubseteq \forall(i : s).\top_i$, $\neg\top_i \sqsubseteq \forall(i : s).\bot_g$

However, in contrast to the approach taken in [4], we will translate bridge rules in DDL as GCIs in GDL. Hence, $\#(\Sigma)$ will include in addition to the above:

- $\#(i : C) \sqsubseteq \#(j : D)$ for all $i : C \xrightarrow{\sqsubseteq} j : D \in B_{ij}$
- $\#(i : C) \sqsupseteq \#(j : D)$ for all $i : C \xrightarrow{\sqsupseteq} j : D \in B_{ij}$

Since the motivation of DDL bridge rules is to simulate concept subsumption as mentioned in DDL proposals [4,5,17], we believe that GCIs offer a more appropriate translation for bridge rules in comparing the result of reasoning in the distributed setting with that of the centralized setting. Note that the semantic difficulties of DDL under incomplete modelling is actually due to the semantic differences between concept subsumptions (i.e., GCIs) and bridge rules (as shown in the Examples 2 and 3). The following theorem reveals that the domain relations being one-to-one is a sufficient condition for exact reasoning in DDL if bridge rules are intended to represent inter-module concept inclusions (proof can be found in appendix).

Theorem 1 *Suppose $\Sigma = \langle\{L_i\}, \{B_{ij}\}\rangle$ is a DDL KB, where none of L_i uses role constants or role constructors, and all domain relations in all models of Σ are one-to-one, then*

- *$\#(\Sigma) \vDash \#(i : X) \sqsubseteq \#(i : Y)$ if and only if $\Sigma \vDash_d i : X \sqsubseteq i : Y$*
- *$\#(\Sigma) \vDash \#(i : X) \sqsubseteq \#(j : Y)$ if and only if $\Sigma \vDash_d (i : X \xrightarrow{\sqsubseteq} j : Y$ or $(j : Y \xrightarrow{\sqsupseteq} i : X)$*

At present, there is no principled approach in DDL to specify such domain relations. Adding $\neg C \xrightarrow{\sqsubseteq} \neg D$ for each $C \xrightarrow{\sqsupseteq} D$, as suggested in [17], does not necessarily result in injective (and hence, also not one-to-one) domain relations for any inter-module concept relations.

Example 4 *A KB Σ_d includes modules $L_{\{1,2\}}$, TBox of L_1 is $\{Woman \equiv \neg Man\}$, TBox of L_2 is $\{Girl \equiv \neg Boy\}$; bridge rules $B_{12} = \{1 : Man \xrightarrow{\sqsupseteq} 2 : Boy\}$. According to* [17], *we should also add $\neg 1 : Man \xrightarrow{\sqsubseteq} \neg 2 : Boy$ i.e.*

$1 : Woman \xrightarrow{\sqsubseteq} 2 : Girl$ to B_{12}. However, that doesn't rule out the possibility of a Girl object being both an image of a Man object and a $Woman$ object, neither ensure one-to-one correspondence between Man objects and Boy objects.

Example 5 *(adopted from* [20]*) Module L_1 entails $\top \sqsubseteq 1 : Car$, module L_2 entails $UsefulThing \sqsubseteq \neg UselessThing$, and there are bridge rules $1 : Car \xrightarrow{\sqsubseteq} 2 : UsefulThing$ and $1 : Car \xrightarrow{\sqsubseteq} 2 : UselessThing$. There is no required new bridge rules to be added according to* [17]*. However, $1 : Car$ is not unsatisfiable, since DDL semantics allows empty domain relations.*

DDL, as presented in [4], meets the *localized semantics* and *directional semantic relations* requirements, but *not* the *exact reasoning* and *transitive reusability* requirements. In general, DDL in its present form does not provide a satisfactory formalism for inter-module or inter-ontology subsumption. In the following text, we will show it can be improved by restricting domain relations to be one-to-one, by P-DL or a combination of DDL and $\mathcal{E}$-connections.

5 Semantics of Linking in $\mathcal{E}$-connections

While DDL allows only one type of domain relations, $\mathcal{E}$-connections allow multiple "link" relations between two domains, such as *worksIn* and *bornIn* between $2 : Person$ and $1 : Region$. Some ideas of $\mathcal{E}$-connection can be traced back to the fusion of abstract description systems (ADS) [1], in which atomic roles are partitioned into disjoint sets that each can only be used in the constructors of the language of a single module. $\mathcal{E}$-connections between ADSs [13], and in particular, between DLs [12,10], restrict the local domains of the $\mathcal{E}$-connected ontology modules to be disjoint. Roles are divided into disjoint sets of *local roles* (connecting concepts in one module) and *links* (connecting inter-module concepts).

Formally, given ontology modules $\{L_i\}$, an (one-way binary) link (more expressive $\mathcal{E}$-connections are beyond the scope of this paper) $E \in \mathcal{E}_{ij}$, where $\mathcal{E}_{ij}, i \neq j$ is the set of all links from the module i to the module j, can be used to construct a concept in module i, with the syntax and semantics specified as follows:

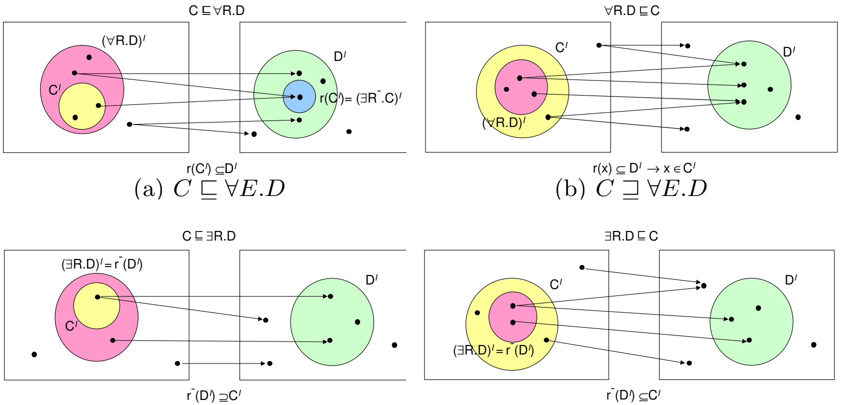
- $\langle E \rangle (j : C)$ or $\exists E.(j : C) : \{x \in \Delta_i | \exists y \in \Delta_j, (x, y) \in E^M, y \in C^M\}$
- $\forall E.(j : C) : \{x \in \Delta_i | \forall y \in \Delta_j, (x, y) \in E^M \to y \in C^M\}\}$
- $\leq nE.(j : C) : \{x \in \Delta_i | \#(\{y \in \Delta_j | (x, y) \in E^M, y \in C^M\}) \leq n\}$
- $\geq nE.(j : C) : \{x \in \Delta_i | \#(\{y \in \Delta_j | (x, y) \in E^M, y \in C^M\}) \geq n\}$

where $M = \langle \{m_i\}, \{E^M\}_{E \in \mathcal{E}_{ij}} \rangle$ is a model of the $\mathcal{E}$-connected ontology, m_i is the local model of L_i; C is a concept in L_j, with interpretation $C^M = C^{m_j}$; $E^M \subseteq \Delta_i \times \Delta_j$ is the interpretation of an $\mathcal{E}$-connection E.

An $\mathcal{E}$-connection model M can be mapped to a DFOL model $M_d = \langle \{m_i\}, \{r_{ij}\} \rangle$ with each E^M ($E \in \mathcal{E}_{ij}$) acting as a domain relation r_{ij} [18]. Extending the semantics of $\mathcal{E}$-connection axioms ((1) and (3) below) given in [18] so as to allow the use of constructed concepts ($\exists E.D$ and $\forall E.D$) on either side of the subsumption, we have (also see Figure 3):

1) $C \sqsubseteq \forall E.D : E^M(C^{m_i}) \subseteq D^{m_j}$
2) $C \sqsupseteq \forall E.D : (\neg C)^{m_i} \subseteq (E^M)^-((\neg D)^{m_j})$, i.e., $\forall x \in \Delta_i, E^M(x) \subseteq D^{m_j} \rightarrow x \in C^{m_i}$
3) $C \sqsubseteq \exists E.D : C^{m_i} \subseteq (E^M)^-(D^{m_j})$
4) $C \sqsupseteq \exists E.D : E^M((\neg C)^{m_i}) \subseteq (\neg D)^{m_j}$, i.e., $C^{m_i} \supseteq (E^M)^-(D^{m_j})$

where $(E^M)^-$ is the inverse of E^M, C is an i-concept and D is a j-concept, C can be an atomic or complex concept. Note that case (2)(similarly also for (4)) can not be reduce to defining $C' \equiv \forall E.D$ and $C' \sqsubseteq C$ in i, since $\equiv$ is the short for $\sqsubseteq$ and $\sqsupseteq$.



(a) $C \sqsubseteq \forall E.D$

(b) $C \sqsupseteq \forall E.D$

(c) $C \sqsubseteq \exists E.D$

(d) $C \sqsupseteq \exists E.D$

r is the interpretation of link E, also acting as the domain relation between the two local domain.

Fig. 3. Semantics of $\mathcal{E}$-Connections

It has been argued that $\mathcal{E}$-connections are more expressive than DDL [13,8] because DDL can be reduced to $\mathcal{E}$-connections. However, the reliance of the reduction on the equivalence of $C \xrightarrow{\sqsubseteq} D$ to $\langle E\rangle.C \sqsubseteq D$ and $C \xrightarrow{\sqsupseteq} D$ to $\langle E\rangle.C \sqsupseteq D$ [13,8], presents semantic difficulties with regard to DDL and $\mathcal{E}$-connections semantics in the DFOL framework [18]: ONTO($\xrightarrow{\sqsupseteq}$) rules in DDL is actually mapped to type d interpretation constraints in DFOL while $\langle E\rangle.C \sqsupseteq D$ is mapped to type b interpretation constraints in DFOL.

We show that *inverse links* being allowed is a necessary condition for $\mathcal{E}$-connections to be more expressive than DDL bridge rules:

Theorem 2 *$\mathcal{E}$-connections, as presented in* [13,8] *is strictly more expressive than DDL as presented in* [4]*, only if inverse links are allowed.*

Proof Sketch: Comparison of the semantics of DDL and $\mathcal{E}$-connections, if we treat the only domain relation in DDL as a $\mathcal{E}$-connection E, as shown in [18,4], $C \sqsubseteq \forall E.D$ has the same semantics as the "into" rule $C \xrightarrow{\sqsubseteq} D$ $(r_{ij}(C^{m_i}) \subseteq D^{m_j})$. However, onto rules, such as $C \xrightarrow{\sqsupseteq} D$ $(r_{ij}(C^{m_i}) \supseteq D^{m_j})$, can be translated into $D \sqsubseteq \exists E^{-}.C$ only if the inversion of $\mathcal{E}$-connections is allowed. □

Thus, the language $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ is more expressive than DDL but $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ (allowing no inverse link) [8] is *not*.

Note that for $i : C \xrightarrow{\sqsupseteq} j : D$, defining an $\mathcal{E}$-connection F from j to i, the onto rule still cannot be translated into $D \sqsubseteq \exists F.C$, since DDL semantics doesn't assume $r_{ij} = r_{ji}^{-}$, therefore $F \neq E^{-}$. To assert the inverse relation, we still need inverse link constructors in $\mathcal{E}$-connections.

Another fundamental difference between the two formalism is the linking subjectivity. In DDL, a bridge rule B_{ij} is always the subjective point of view of j and to be used by reasoning in j. Nevertheless, a $\mathcal{E}$-connection $\mathcal{E}_{ij}$ provides the module i the ability to construct new concepts therefore actually represents i's point of view of the domain relation, and is used by reasoning in i. It can be confirmed in the syntax proposal of $\mathcal{E}$-connections [8] where the source of a link is the module in which it has been declared.

$\mathcal{E}$-connections allow multiple links between modules and the construction of new concepts e.g. $WorkForce \equiv \langle worksIn \rangle Region$, while DDL does not. Module transitive useability can be realized in a limited form by transitive links [14]. $\mathcal{E}$-connections are also directional. Reasoning in $\mathcal{E}$-connections without generalized links is exact w.r.t a combined TBox of the $\mathcal{E}$-connected ontology, since a concept is satisfiable in the $\mathcal{E}$-connected ontology if and only if there is a combined model for the combined TBox and the concept [10,9].

However, the applicability of $\mathcal{E}$-connections in practice is also limited by the need to ensure that the local domains are disjoint:

- To enforce local domain disjointness, a concept cannot be declared as subclass of another concept in a foreign module thereby ruling out the possibility of asserting inter-module subsumption and the general support for transitive useability; a property cannot be declared as sub-relation of a foreign property; neither foreign classes nor foreign properties can be instantiated; cross-module concept conjunction or disjunction are also illegal.
- $\mathcal{E}$-connected ontologies have difficulties to be used with OWL importing mechanism, since importing may actually "decouple" the combination and result in inconsistency [8].
- $\mathcal{E}$-connected ontologies do not allow a same term be used as both a link name and a local role name, nor role inclusions between links and roles, while such features are widely required in practice [8]. The"punning" approach [8], where a same name can have different interpretations, is rather as a syntactical sugar than a semantic solution to such problems.

6 Semantics of Importing – OWL

Our investigation on the semantics of linking reveals that, such approaches, while established the critical initial steps towards modular ontologies and combination of multiple ontologies, cannot provide a general and semantically sound framework for many urgently required inference tasks, such as transitive inter-module subsumption. Such a observation suggests that, many of the semantic difficulties of linking approaches are the result of the local language and local model disjointness assumption. Consequently, the question is, if it is generally necessary to require such disjointness, or what is the best tradeoff between module independence and the expressivity/reasoning power of the modular ontology?

OWL [16] does not make such module disjointness assumption and adopts an *importing* mechanism to support integration of ontology modules. An OWL ontology may contain `owl:imports` annotations with references to other OWL ontologies. Once an OWL ontology O_1 imports another OWL ontology O_2, the terms defined in O_2 can be directly used in O_1 as *foreign terms.* In this way, different ontologies can be connected into a larger ontology containing all terms and axioms in all constituting ontologies, whereas a whole ontology can also be divided into smaller components within separate identification spaces, such as XML name spaces.

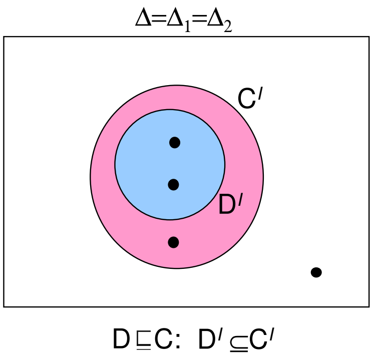
However, the importing mechanism in OWL, in its current form, suffers from several serious drawbacks:

First, it doesn't support *localized semantics.* The OWL semantics [15] requires for any OWL ontology O and an abstract OWL interpretation I of O, "I satisfies each ontology mentioned in an owl:imports annotation directive of O". Therefore, it directly introduces both terms and axioms of the imported ontologies into the importing ontology, which results in a global interpretation of all the ontology modules [4]. In other words, the models of all ontology modules are completely overlapping into a single global model (as the example in Figure 4 shows). Reasoning with `owl:imports`, although is exact to the merged ontology of all recursively imported ontologies, is not distributed but centralized.

Second, importing in OWL is actually not *directional semantic relation.* Since there is a required global model, the semantic constraints specified in a referring ontology will also be completely required over the imported ontologies, i.e. there may be information "backflow".

Third, it provides no support for *partial reuse* of an ontology module. An ontology has to be completely imported or be completely discarded therein precludes the possibility of partially reusing. By recursive importing, an ontology may import large chunks of unneeded knowledge from the imported ones.

In short, OWL importing mechanism at present form [16] is only a syntactic solution for modular ontologies, and cannot provide a sound modular semantics for a distributed knowledge base on the web.



Example of the global model of an ontology with two modules using the OWL importing mechanism. Concept C is defined in module 1, Concept D is defined in module 2. Module 2 imports module 1 and declares $2 : D \sqsubseteq 1 : C$. Any model of each module is identical to the global model $\mathcal{I}$, with same interpretation domain Δ. $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ is required for the whole ontology.

Fig. 4. Example of OWL ontology Semantics

7 Semantics of Importing – P-DL

Package-based Description Logics (P-DL)[3] offer a tradeoff between the strong module disjointness assumption of DDL and $\mathcal{E}$-connections, and the OWL importing mechanics, which forces *complete overlapping* of modules.

7.1 Syntax and Semantics of P-DL

In P-DL, an ontology is composed of a collection of modules called *packages*. Each term (name of a concept, a property or an individual) and each axiom is associated with a *home package*. A package can use terms defined in other packages i.e., *foreign terms*. If a package L_j uses a term $i : t$ with home package L_i $(i \neq j)$, then we say t is *imported* into L_j, and the importing relation is denoted as r^t_{ij}. L_i may contain the usual TBox and ABox of DL. For simplicity, we do not present advanced features of P-DL, such as package hierarchy and scope limitation modifiers [3].

We denote the package extension to DL as $\mathcal{P}$. For example, $\mathcal{ALCP}$ is the package-based version of DL $\mathcal{ALC}$. In what follows, we will examine a restricted type of package extension which only allows import of concept names, denoted as $\mathcal{P}_{\mathcal{C}}$. We will show that even this restricted form of package extension is not trivial and is more expressive than DDL and $\mathcal{E}$-connection.

The semantics of P-DL is expressed in DFOL as follows: For a package-based ontology $\langle \{L_i\}, \{r^t_{ij}\}_{i \neq j} \rangle$, a distributed model is $M = \langle \{m_i\}, \{(r^t_{ij})^M\}_{i \neq j} \rangle$, where m_i is the local model of module i, $(r^t_{ij})^M \subseteq \Delta_i \times \Delta_j$ is the interpretation for the importing relation r^t_{ij}, which meets the following requirements:

– Every importing relation is one-to-one, and for every object in t^{m_j} there is a single unique object in t^{m_i} as its pre-image (therefore $r_{ij}(t^{m_i}) = t^{m_j}$).
– Term Consistency: importing relations are consistent for different terms. Each object in the model of a source package corresponds uniquely to an object in the model of any target package for interpretations of importing relations of all terms, i.e., for any $i : t_1 \neq i : t_2$ and any $x, x_1, x_2 \in \Delta_i$, $(r_{ij}^{t_1})^M(x) = (r_{ij}^{t_2})^M(x)$ and $(r_{ij}^{t_1})^M(x_1) = (r_{ij}^{t_2})^M(x_2) \neq \emptyset \rightarrow x_1 = x_2$.
– Compositional Consistency: if $(r_{ik}^{i:t_1})^M(x) = y_1$, $(r_{ij}^{i:t_2})^M(x) = y_2$, $(r_{jk}^{j:t_3})^M(y_2) = y_3$, , (where t_1 and t_2 may or may not be same), and y_1, y_2, y_3 are not null, then $y_1 = y_3$. Compositional consistency helps ensure that the transitive reusability property holds for P-DL.

The domain relation between m_i and m_j is $r_{ij} = \cup_t (r_{ij}^t)^M$.

Lemma 4 *Domain relations in a P-DL model are one-to-one.*

Lemma 4 states that a domain relation r_{ij} in a P-DL model isomorphically maps, or copies, the relevant partial domain from m_i to m_j. For any concept $i : C$, $r_{ij}(C^{m_i})$, if not empty, contains the copy of a subset of objects in C^{m_i}. Such domain relations allow us to relax the domain disjointness assumption adopted in DDL and $\mathcal{E}$-connections, since the construction of a local model is partially dependent on the structure of local models of imported modules, with the benefits of preserving exact semantics of terms shared by different modules.

Immediately from the one-to-one domain relation property, we have:

Lemma 5 *In a P-DL model m, for any domain relation r_{ij} and concept $i : C$, we have $r_{ij}(C^{m_i}) \cap r_{ij}((\neg C)^{m_i}) = \emptyset$.*

If a term is indirectly used in a non-home package, compositional consistency property of domain relations makes the domain relation inferrable. For example, if some terms defined in L_1 are imported into L_2, and some terms in L_2 are imported into L_3, then the importing relation r_{13} can be inferred from the composition $r_{12} \circ r_{23}$.

Lemma 6 *For domain relations in a model of P-DL, $r_{ik} = r_{ij} \circ r_{jk}$, $i \neq j, j \neq k$.*

In the following text, r_{ij} refers to either an explicitly given domain relation or an inferred domain relation, or their union, between package i and j.

If $i : C$ is imported into j, we define inter-module subsumption $i : C \sqsubseteq_P j : D$ as $r_{ij}(C^{m_i}) = C^{m_j} \subseteq D^{m_j}$ and $i : C \sqsupseteq_P j : D$ as $r_{ij}(C^{m_i}) = C^{m_j} \supseteq D^{m_j}$ (see Figure 5). Note that inter-module subsumption is substantially different from bridge rules in DDL. DDL bridge rules bridge semantic gaps between *different* concepts, and there is no principled way to ensure subjective domain relations to be semantically consistent (in the sense of one-to-one mappings and compositional consistency). In contrast, P-DL importing mechanism bridges the semantic gaps between multiple references of the *same* concept in different modules. Importing of C from i to j cannot be reduced to a DDL equivalency bridge rule

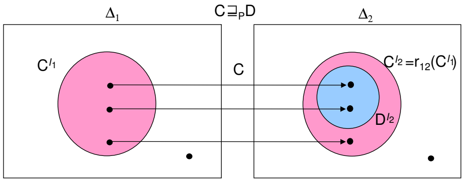


Fig. 5. Semantics of P-DL

$C \overset{\equiv}{\longrightarrow} C'$, since in DDL $r_{ij}(C^{m_i}) = C'^{m_j}$ does not guarantee C^{m_i} and C'^{m_j} are interpretations for the same concept.

We show below that such a relaxation of module disjointness does not sacrifice localized semantics and can help us to solve many semantic difficulties presented in other approaches and provide stronger expressivity.

7.2 Features of P-DL Semantics

The loss of local model disjointness in P-DL does not sacrifice *localized semantics* property of modules, since the local models (unlike in OWL which requires *completely* overlapping of local models) are, only *partially* overlapping. The semantics of the part of a module that is not exported to any other module remains local to that module. Consequently, there is no required global model. The example below demonstrates that P-DL also satisfies *directional semantic relation* and module *transitive reusability* properties.

Example 6 *Consider four modules $L_{\{1,2,3,4\}}$ as shown in Figure 6.*

1. *Transitivity of inter-module subsumption holds: $r_{14}(A^{m_1}) = r_{24}(r_{12}(A^{m_1})) \subseteq r_{24}(r_{12}(B^{m_1})) = r_{24}(B^{m_2}) \subseteq r_{24}(C^{m_2}) \subseteq r_{24}(P^{m_2}) = P^{m_4} \subseteq Q^{m_4}$, i.e., $A \sqsubseteq_P Q$. Although no term in L_1 is directly imported into L_4, we can infer the domain relation r_{14} from $r_{12j} \circ r_{24}$ utilizing their compositional consistency property.*
2. *The importing relation is directional. Thus, $r_{12}(A^{m_1}) \subseteq r_{12}(D^{m_1})$ is enforced only in L_2, while $A^{m_1} \subseteq D^{m_1}$ is not required in L_1. There is no information "backflow" in importing. Therefore, while L_2 and L_3 are inconsistent, they are all consistent to L_1, and the consistency of L_1 is still guaranteed.*
3. *The model overlapping is only partial, e.g., E and F in* 1 *are semantically separated from L_2 and have no correspondence in the local model m_2.*

An integrated KB can be obtained from a P-DL KB by combining axioms in all packages. Because of the isomorphic nature of importing relations, we have the theorem:

Theorem 3 *Reasoning in a P-DL KB is exact w.r.t. its integrated KB.*

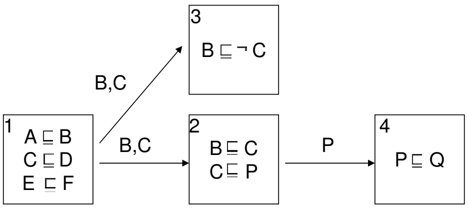


Fig. 6. P-DL Ontology Example

The intuition behind this proof is as follows: since the only type of semantic relations between modules are importing relations, and shared terms are always interpreted consistently in different modules, we can transform a distributed P-DL model into a classic DL model by merging all "copied" objects in each of the local models.

However, a limitation of the importing approach adopted by P-DL is that the general decidability transfer property does not always hold in P-DL since the union of two decidable fragments of DL may not in general be decidable [1]. This presents semantic difficulties in the general setting of connecting ADSs [1,13]. Fortunately, in the case of a web ontology language where different ontology modules are specified using subsets of the *same* decidable DL language such as $\mathcal{SHOIQ}(D)$ (OWL-DL), the union of such modules is decidable. With the availability of the decision procedure [11] and highly optimized reasoners for $\mathcal{SHOIQ}(D)$, we can ensure the decidability of P-DL ontology within the modular web ontology setting. Therefore we have:

Theorem 4 *$\mathcal{SHOIQP_C}(D)$ is decidable.*

P-DL also has some expressivity limitations. $\mathcal{P_C}$ does not allow role inclusions across different modules, using foreign role to construct local concept, declaring a local role as the inverse of a foreign role, nor the importing of nominals defined in other modules. Due to the one-to-one domain relation requirement, P-DL can support only one-to-one ABox mappings, and not many-to-one, one-to-many, or many-to-many ABox mappings. The semantics of more expressive P-DL that allows importing of role and individual names still needs further investigation.

7.3 Relation between P-DL, DDL and $\mathcal{E}$-Connections

P-DL, despite its stronger domain relation restrictions, can be used to model DDLs and $\mathcal{E}$-Connections.

The reduction from DDL to P-DL is straightforward. An into rule $i : C \xrightarrow{\sqsubseteq} j : D$ in DDL can be reduced to a P-DL axiom $C \sqsubseteq_P D$ in module j and C is an imported concept; similarly, an onto rule $i : C \xrightarrow{\sqsupseteq} j : D$ in DDL is translated as $C \sqsupseteq_P D$ in module j and C is an imported concept. The semantic interpretation of such a reduction is clear since $r_{ij}(C^{m_i}) = C^{m_j}$, therefore $r_{ij}(C^{m_i}) \subseteq D^{m_j}$ iff $C^{m_j} \subseteq D^{m_j}$ and $r_{ij}(C^{m_i}) \supseteq D^{m_j}$ iff $C^{m_j} \supseteq D^{m_j}$.

P-DL may avoid the semantic difficulties presented in DDL.

Example 7 (Subsumption Propagation) *A P-DL KB includes three modules $L_{\{1,2,3\}}$, L_1 has empty TBox, $L_2 = \{1 : Bird \sqsupseteq 2 : Fowl\}$, $L_3 = \{2 : Fowl \sqsupseteq 3 : Chicken\}$, importing relations are $r_{12}^{Bird}, r_{23}^{Fowl}$. The inter-module subsumption problem $1 : Bird \sqsupseteq 3 : Chicken$ can be answered, since in any model of the KB, $Chicken^{m_3} \subseteq r_{23}(Fowl^{m_2}) \subseteq r_{23}(r_{12}(Bird^{m_1})) = r_{13}(Bird^{m_1})$, therefore $Bird \sqsupseteq_P Chicken$ is always true.*

Example 8 (Inter-module Unsatisfiability) *A P-DL ontology contains two modules $L_{\{1,2\}}$, $L_1 = \{1 : Bird \sqsubseteq 1 : Fly\}$, $L_2 = \{1 : Bird \sqsupseteq 2 : Penguin, \neg 1 : Fly \sqsupseteq 2 : Penguin\}$, importing relations are $r_{12}^{Bird}, r_{12}^{Fly}$. For any model m of the ontology, $Penguin^{m_2} \subseteq r_{12}((\neg Fly)^{m_1}) \cap r_{12}(Bird^{m_1})$, and $r_{12}(Bird^{m_1}) \subseteq r_{12}(Fly^{m_1})$. Therefore $Penguin^{m_2} \subseteq r_{12}((\neg Fly)^{m_1}) \cap r_{12}(Fly^{m_1}) \subseteq \emptyset$ is not satisfiable.*

An $\mathcal{E}$-connection-like constructed concept such as $\exists(i : E).(j : D)$ can be defined in the module i, where $j : D$ is imported into i, with semantics: $\{x \in \Delta_i | \exists y \in \Delta_j, y' = r_{ji}(y) \in \Delta_i, (x, y') \in E^{m_i}, y \in D^{m_j}\}\}$. $\forall(i : E).(j : D)$ can be constructed similarly. It is easy to see a combined model (Tableau) of $\mathcal{E}$-connections [8] can be reduced to a P-DL model by transforming every $\mathcal{E}$-connection instance $e(i : x, j : y)$ to a role instance $e(i : x, i : y')$ and adding (y, y') to the domain relation r_{ji} if it has not already been added.

Since "links" in $\mathcal{E}$-connections can be specified as local roles in P-DL with foreign concepts as ranges, link inclusion, link inverse, and link number restriction can also be reduced into normal role axioms in P-DL. Therefore, we have:

Theorem 5 *P-DL $\mathcal{SHOIQP_C}(D)$ is strictly more expressive than the DDL extension to $\mathcal{SHOIQ}$ with bridge rules between concepts, and $\mathcal{E}$-Connections $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}}$ $(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ and $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$.*

Some types of DDL and $\mathcal{E}$-Connections can not be reduced to the P-DL extension $\mathcal{P_C}$, e.g., DDL bridge rules between roles and individuals or generalized links [14] in $\mathcal{E}$-connections. However, we believe future extension of P-DL may cover some of these scenarios.

Another observation is that it is possible to simulate the one-to-one domain relations that are required in P-DL by the combination of DDL and E-connections[1]. If we use bridge rules as a special type of $\mathcal{E}$-connections with "≤ 1" cardinality restriction in $\mathcal{E}$-connections, it effectively encodes the one-to-one domain relations. More precisely, for any pair of module i, j, if we denote E as the $\mathcal{E}$-connection for bridge rules from i to j, F as the $\mathcal{E}$-connection for bridge rules from j to i,, the following axioms can be added:

- In module i: $\top_i \sqsubseteq \leq 1E.\top_j$
- In module j: $\top_j \sqsubseteq \leq 1F.\top_i$
- $F = E^-$.

[1] We thank the anonymous reviewers of the Description Logics Workshop for pointing this out.

However, such a simulation does not always meet the compositional consistency requirement of P-DL. Therefore, such a combination of DDL and $\mathcal{E}$-connections, while it can solve the inter-module unsatisfiability problem, may fail on some problems that require module transitive reusability, such as the general subsumption propagation problem as outlined in Example 2.

8 Summary

In this paper, we have investigated the semantics of DDL, $\mathcal{E}$-connections and P-DL. We have shown that (a) one-to-one domain relation is a sufficient condition for exact DDL reasoning; (b) $\mathcal{E}$-connections, in general, are more expressive than DDL only with inverse links; c) an importing approach in P-DL can be used to ensure transitivity of inter-module subsumption without sacrificing the exactness of inference in P-DL with only a compromise of local semantics. Our results raise the possibility of avoiding many of the semantic difficulties in current modular ontology language proposals by removing the strong assumption of module disjointness.

Acknowledgements: This research was supported in part by grants from the US NSF (0219699, 0639230) and NIH (GM 066387).

Appendix: Proof of Theorem 1

Given a DDL KB Σ and its integration $\#(\Sigma)$ as outlined in Section 4, we first prove that distributed models of Σ and classic models of $\#(\Sigma)$ can be mapped to each other.

For any distributed model $M_d = \langle\{\mathcal{I}_i\}, \{r_{ij}\}\rangle$ of Σ, where $\mathcal{I}_i = \langle\Delta_i, (.)^{\mathcal{I}_i}\rangle$, we can construct a classic model $\mathcal{I} = \langle\Delta^{\mathcal{I}}, (.)^{\mathcal{I}}\rangle$ by merging individual pairs connected by domain relations into single individuals in $\mathcal{I}$. For any concept or role $i : A$, $(i : A)^{\mathcal{I}}$ in $\mathcal{I}$ is the same as $A^{\mathcal{I}_i}$ in M_d. It is easy to see that

- $A^{\mathcal{I}_i} \subseteq B^{\mathcal{I}_i} \to (i : A)^{\mathcal{I}} \subseteq (i : B)^{\mathcal{I}}$
- $\Delta_i \subseteq \Delta^{\mathcal{I}}$
- $(i : A)^{\mathcal{I}} \subseteq \Delta_i$ for every atomic concept $i : A$
- for any $\langle x, y\rangle \in (i : s)^{\mathcal{I}}$, we have $x \in \Delta_i$ and $y \in \Delta_i$
- (INTO) $r_{ij}(C^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_i} \to (i : C)^{\mathcal{I}} \subseteq (i : D)^{\mathcal{I}}$
- (ONTO) $r_{ij}(C^{\mathcal{I}_i}) \supseteq D^{\mathcal{I}_i} \to (i : C)^{\mathcal{I}} \supseteq (i : D)^{\mathcal{I}}$

The last two conditions are true since domain relations r_{ij} in M_d are one-to-one. Therefore, $\mathcal{I}$ is a model of $\#(\Sigma)$.

In the other direction, given a classic model $\mathcal{I} = \langle\Delta^{\mathcal{I}}, (.)^{\mathcal{I}}\rangle$ of the integrated KB $\#(\Sigma)$, we may also construct a distributed model $M_d = \langle\{\mathcal{I}_i\}, \{r_{ij}\}\rangle$ (where $\mathcal{I}_i = \langle\Delta_i, (.)^{\mathcal{I}_i}\rangle$) of Σ in the following way: for any concept name $i : A$, Δ_i contains all individuals of $(i : A)^{\mathcal{I}}$. $(.)^{\mathcal{I}_i}$ maps each i-concept and i-role to the same set as $(.)^{\mathcal{I}}$ does. For any bridge rule $C \xrightarrow{\sqsubseteq} D$ or $C \xrightarrow{\sqsupseteq} D$ in B_{ij}, Δ_j also contains all individuals of $(i : C)^{\mathcal{I}}$ and the domain relation r_{ij} contains shared individual pairs in Δ_i and Δ_j.

It is easy to see that by making "copies" of some individuals in different local domains such a decomposition creates a distributed model for Σ. The domain relations in the distributed model are one-to-one.

Now we prove reasoning in Σ is exact w.r.t the integrated ontology $\#(\Sigma)$. For any i-concept $i : X$ and $i : Y$, if $\Sigma \models_d i : X \sqsubseteq i : Y$, we have that for any distributed model M_d of Σ, $X^{\mathcal{I}_i} \subseteq Y^{\mathcal{I}_i}$. Consequently, since any classic model $\mathcal{I}$ of $\#(\Sigma)$ can be reduced to a distributed model, $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$ must be hold, i.e. $\#(\Sigma) \models \#(i : X) \sqsubseteq \#(i : Y)$. Similarly, if $\#(\Sigma) \models \#(i : X) \sqsubseteq \#(i : Y)$, we also have $\Sigma \models_d i : X \sqsubseteq i : Y$.

For any i-concept $i : X$ and j-concept $j : Y$, if $\Sigma \models_d i : X \xrightarrow{\sqsubseteq} j : Y$, for any distributed model M_d of Σ, we have: $r_{ij}(X^{\mathcal{I}_i}) \subseteq Y^{\mathcal{I}_j}$. Consequently, for any classic model $\mathcal{I}$ of $\#(\Sigma)$ reduced from M_d, since r_{ij} is one-to-one, $r_{ij}(X^{\mathcal{I}_i})$ will be merged with $X^{\mathcal{I}_i}$. Hence, $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$ in any $\mathcal{I}$, i.e., $\#(\Sigma) \models \#(i : X) \sqsubseteq \#(j : Y)$. Similarly, if $\Sigma \models_d i : X \xrightarrow{\sqsupseteq} j : Y$, we will have $\#(\Sigma) \models \#(i : X) \sqsupseteq \#(j : Y)$.

For any i-concept $i : X$ and j-concept $j : Y$, if $\#(\Sigma) \models \#(i : X) \sqsubseteq \#(j : Y)$, for any classical model $\mathcal{I}$ of $\#(\Sigma)$, we have: $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. For any distributed model M_d that is obtained from $\mathcal{I}$, either $r_{ij}(X^{\mathcal{I}_i}) \subseteq Y^{\mathcal{I}_j}$ or $r_{ji}(Y^{\mathcal{I}_j}) \supseteq X^{\mathcal{I}_i}$, i.e., $\Sigma \models_d (X \xrightarrow{\sqsubseteq} Y$ or $Y \xrightarrow{\sqsupseteq} X)$.

Therefore, if domain relations in DDL are one-to-one, the results of reasoning in DDL are identical to the results of reasoning with the integrated ontology.

Proof of Theorem 3

First, we need to construct a centralized (classic) KB given a P-DL KB. It can be accomplished by merging imported terms. Formally, A reduction $\Re$ from a P-DL KB $\Sigma_d = \langle \{L_i\}, \{r^t_{ij}\}_{i \neq j} \rangle$ to a classic KB Σ can be done in the following way:

- Σ contains all local terms (concept, property, individual names), not including any (imported) foreign terms, of each package L_i.
- Copy all constant concepts $\top_i$(anything) and $\bot_i$(nothing), for all i, into Σ; add the global anything $\top$ and global nothing $\bot$ into Σ; add $\bot_i \sqsubseteq \bot$ to the TBox of Σ for each i.
- Copy each axiom $j : C \sqsubseteq k : D$ in the TBox of L_i as $\#(j : C) \sqsubseteq \#(k : D)$ into the TBox of Σ, j, k may be or not be the same of i, and j, k may or may not be the same. $\#()$ is defined as below.
- For each local atomic concept C in L_i, add $i : C \sqsubseteq \top_i$ in the TBox of Σ, for all i.
- For each local atomic role P in L_i, add $\top_i$ as its domain, i.e. add $\top \sqsubseteq \forall P^-.\top_i$ in the TBox of Σ, for all i.
- For each importing relation $r^{i:C}_{ij}$, add $i : C \sqsubseteq \top_j$ in the TBox of Σ, for all i, j.

- Copy all assertions $j : C(a)$ in the ABox of L_i, j may be or not be i, as $\#(j : C)(a)$ into the ABox of Σ, for all i.
- Copy all assertions $i : p(i : a, i : b)$ in the ABox of L_i, as $\#(i : p)(a, b)$ into the ABox of Σ, for all i.

The mapping $\#()$ is defined with modifications to enable imported terms from its original form in [4] as: for atomic concepts, roles, and individuals $i : M$ used in module j (i may be or not be j), $\#(j : (i : M)) = i : M \sqcap \top_i \sqcap \top_j$; for a complex concept constructor ρ with k arguments, $\#(i : \rho(X_1, ..., X_k)) = \top_i \sqcap \rho(\#(X_1), ..., \#(X_k))$. For example $\#(i : (\neg j : C)) = \top_i \sqcap (\top_j \sqcap \neg j : C)$, $\#i : (\forall i : p.j : C) = \top_i \sqcap \forall(i : p).(\top_i \sqcap \top_j \sqcap j : C)$.

Since the only type of semantic relations between modules are importing relations, and shared terms are always interpreted consistently in different modules, we can transform a distributed P-DL model into a classic DL model by merging all "copied" objects in each of the local models. Formally, for a model $m_d = \langle \{m_i\}, \{r_{ij}\}_{i \neq j} \rangle$ of a P-DL KB $\Sigma_d = \langle \{L_i\}, \{r^t_{ij}\}_{i \neq j} \rangle$, , where $m_i = \langle \Delta_i, (.)^{m_i} \rangle$ is the local model for L_i, a classic model $m = \langle \Delta, (.)^m \rangle$ of $\Re(\Sigma_d)$ can be constructed in the following way:

1. Δ is the union of all Δ_i: $\Delta = \cup_i \Delta_i$.
2. $(.)^m$ is the union of all $(.)^{m_i}$: For any concept, role, or individual T, $T^m = \cup_i T^{m_i}$.
3. Remove duplicated objects: For any $(x, y) \in r^t_{ij}$, where $x \in \Delta_i, y \in \Delta_j$, remove y from Δ, and change all occurrences of y in $(.)^m$ with x.

Such a reduction, as an example showing in Figure 7, ensures that $m_d \vDash i : (C \sqsubseteq D)$ if and only if $m \vDash (\#(i : (C)) \sqsubseteq \#(i : (D)))$. For any atomic or complex concept C occurring in L_i, C is satisfiable in m_d if and only if $\#(C)$ is satisfiable in m. In general, $m_d \vDash \Sigma_d$ if and only if $m \vDash \Re(\Sigma_d)$. Therefore, any satisfiability or entailment problem in P-DL is exact relative to its centralized counterpart.

References

1. F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics. In *Description Logics*, pages 21–30, 2000.
2. J. Bao, D. Caragea, and V. Honavar. Modular ontologies - a formal investigation of semantics and expressivity. In *R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): Asian Semantic Web Conference 2006, LNCS 4185*, pages 616–631, 2006.
3. J. Bao, D. Caragea, and V. Honavar. Towards collaborative environments for ontology construction and sharing. In *International Symposium on Collaborative Technologies and Systems (CTS 2006)*, pages 99–108. IEEE Press, 2006.
4. A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In *CoopIS*, pages 36–53, 2002.
5. P. Bouquet, F. Giunchiglia, and F. van Harmelen. C-OWL: Contextualizing ontologies. In *Second International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer Verlag, 2003.
6. C. Ghidini and F. Giunchiglia. Local model semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

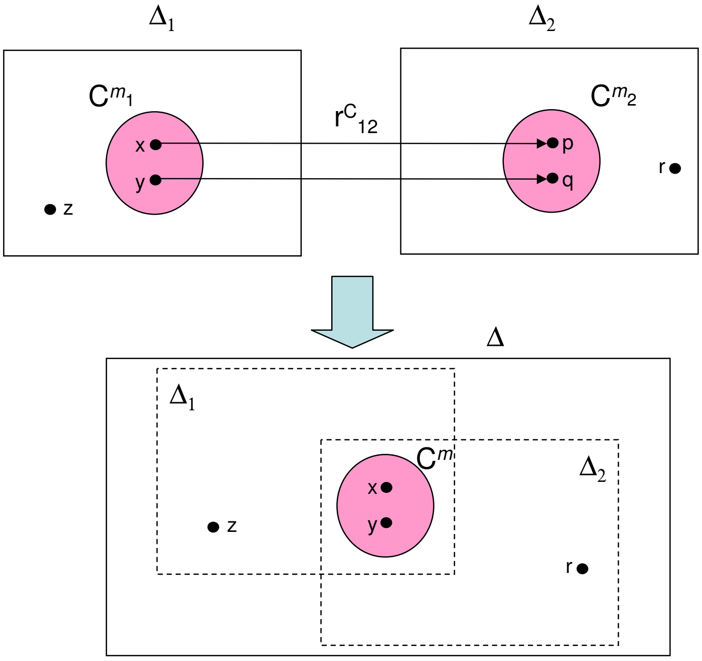


Fig. 7. An Example of Reduction from a P-DL model to a DL model

7. C. Ghidini and L. Serafini. *Frontiers Of Combining Systems 2, Studies in Logic and Computation*, chapter Distributed First Order Logics, pages 121–140. Research Studies Press, 1998.
8. B. C. Grau. *Combination and Integration of Ontologies on the Semantic Web.* PhD thesis, Dpto. de Informatica, Universitat de Valencia, Spain, 2005.
9. B. C. Grau, B. Parsia, and E. Sirin. Tableau algorithms for e-connections of description logics. Technical report, University of Maryland Institute for Advanced Computer Studies (UMIACS), TR 2004-72, 2004.
10. B. C. Grau, B. Parsia, and E. Sirin. Working with multiple ontologies on the semantic web. In *International Semantic Web Conference*, pages 620–634, 2004.
11. I. Horrocks and U. Sattler. A Tableaux Decision Procedure for SHOIQ. In *IJCAI*, pages 448–453, 2005.
12. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of description logics. In *Description Logics Workshop, CEUR-WS Vol 81*, 2003.
13. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.
14. B. Parsia and B. C. Grau. Generalized link properties for expressive epsilon-connections of description logics. In *AAAI*, pages 657–662, 2005.
15. P. Patel-Schneider, P.Hayes, and I. Horrocks. Web ontlogy language (owl) abstract syntax and semantics. http://www.w3.org/TR/owl-semantics/, February 2003.
16. G. Schreiber and M. Dean. Owl web ontology language reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/, February 2004.
17. L. Serafini, A. Borgida, and A. Tamilin. Aspects of distributed and modular ontology reasoning. In *IJCAI*, pages 570–575, 2005.

18. L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In *IJCAI*, pages 576–581, 2005.
19. L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376, 2005.
20. H. Stuckenschmidt, L. Serafini, and H. Wache. Reasoning about ontology mappings. Technical report, Department for Mathematics and Computer Science, University of Mannheim ; TR-2005-011, 2005.