Modular Ontologies - A Formal Investigation of Semantics and Expressivity

Jie Bao, Doina Caragea, and Vasant G. Honavar

Artificial Intelligence Research Laboratory,
Department of Computer Science
Iowa State University, Ames, IA 50011-1040, USA
{baojie, dcaragea, honavar}@cs.iastate.edu

Abstract. With the growing interest in modular ontology languages to address the need for collaborative development, integration, and use of ontologies on the Web, there is an urgent need for a common framework for comparing modular ontology language proposals on the basis of criteria such as their semantic soundness and expressive power. We introduce an Abstract Modular Ontology (AMO) language and offer precise definitions of semantic soundness such as localized semantics and exact reasoning, and expressivity requirements for modular ontology languages. We compare Distributed Description Logics (DDL), $\mathcal{E}$-connections, and Package-Based Description Logics (P-DL) with respect to these criteria. Our analysis suggests that by relaxing the strong domain disjointedness assumption adopted in DDL and $\mathcal{E}$-connection, as P-DL demonstrated, it is possible to overcome some known semantic difficulties and expressivity limitations of DDL and $\mathcal{E}$-Connections.

1 Introduction

Recently, there is a growing interest in modular ontology languages such as Distributed Description Logics (DDL) [4] and its syntax C-OWL[5], $\mathcal{E}$-connections [12,9], Fusion of Abstract Description Systems (FADS) [1], and Package-extended Description Logics (P-DL) [3]. Two broad classes of approaches are adopted to asserting and using semantic relations between multiple ontology modules: DDL and $\mathcal{E}$-connections adopt the "linking" approach that assumes that the modules are *nonoverlapping* or *disjoint*, while P-DL adopts the "importing" approach that allows direct use of foreign terms in an ontology module. Both DDL and P-DL cover scenarios that require inter-module concept subsumptions (e.g., *Dog* is *Animal*), while $\mathcal{E}$-connections allows only inter-module role relations (e.g., *DogOwner* owns *Dog*). Serafini *et.al.* (2005) [16] compared several mapping languages such as DDL and $\mathcal{E}$-connections, by reducing them to the Distributed First Order Logics (DFOL) [7] framework. Others have noted some of the semantic difficulties and limitations of such approaches [9,2].

However, there has been relatively little work on precise requirements for, and criteria for evaluating, modular ontology languages in more general settings that encompass both linking and importing among ontology modules. Some natural questions that arise in comparing different approaches to integration of ontology modules are: What are the minimal requirements for ensuring the semantic soundness of a modular ontology language? What ontology language features are needed to construct a practical modular ontology? Under what circumstances can a reasoning process in a modular ontology language be said to be sound and complete? What are the sources of semantic difficulties in some modular ontology languages? How can such difficulties be avoided?

The goal of this paper is to provide some preliminary answers to these questions. Section 2 points out limitations of OWL to motivate the need for modular ontology languages. Section 3 explores a set of evaluation criteria for modular ontology languages. Section 4 precisely defines the aforementioned criteria within the Abstract Modular Ontology framework. Sections 5 and 6 (respectively) compare the semantic soundness and expressivity of several existing modular ontology language proposals w.r.t. the introduced criteria. 7 concludes with a summary and a brief discussion of related research.

2 Limitations of OWL as a Modular Ontology Language

OWL [15] is among the leading candidates for for a web ontology language. Hence, it is natural to ask why OWL cannot be used as a satisfactory modular ontology language.

We start by observing that OWL adopts an *importing* mechanism to support the integration of ontology modules. Thus, an OWL ontology may contain annotations `owl:imports` 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 manner, an ontology can 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. In what follows, we will illustrate these drawbacks using a concrete example, the well-known Wine Ontology.

The wine ontology is given in two OWL files[1] focused on wine knowledge and general food knowledge, respectively. However, such a division into different files, a.k.a., XML name spaces:

- Does not support *localized semantics*. The inference is necessarily performed on the integrated centralized ontology of Wine and Food. The OWL semantics [14] requires that for any OWL ontology O and any 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 referring ontology (e.g., Food to Wine), which results in a global interpretation of all modules [4].
- Does not allow *local point of view*. All modules are required to adopt completely the same semantic perspective. For example, if the Food module as-

[1] http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine.rdf and http://www.w3.org/TR/2004/REC-owl-guide-20040210/food.rdf

serts "*Meat* and *Fowl* are disjoint", the Wine module cannot adopt another point of view asserting that "*Fowl* is a type of *Meat*".

- Does not support *directional semantic relations.* Since a global model is used, the semantic constraints specified in a referring ontology (e.g., Wine) will be completely transfered over the imported ontologies (e.g., Food). If the Wine module adds "*Fowl* is a type of *Meat*", the consistency of the Food module is also violated.
- Does not support *partial reuse.* Two modules mutually import each other, therefore a user has to import ALL the Wine and Food modules, even though only a small part (such as *Grape* classification) may be needed. Thus in order for an ontology module to be reused, it has to be either completely imported or completely discarded.

In conclusion, the current OWL importing mechanism [14] is only a syntactic solution, not a semantically sound solution for modular web ontologies.

3 Desiderata of Modular Ontologies

The observations in the previous section provide some intuitions that suggest evaluation criteria for modular ontology languages. The first set of criteria that we consider is aimed at evaluating the semantic soundness of such languages:

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 appropriate *integration* of the relevant ontology modules.
3. **Directional Semantic Relations.** The language 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 Wine reuses module Food, and module Food reuses module Drink, then effectively, module Food reuses module Drink. If that is not the case, the knowledge in Food may be used in an unsafe or altered way. For example, if the Drink module contains "*Alcohol* is *Beverage*", the Food module contains "*Beer* is *Alcohol*" and "*Beverage* is *EdibleThing*", the Wine module may not be able to infer "*Beer* is *EdibleThing*" if knowledge in the Drink module is not considered.
5. **Decidability.** The ontology should be decidable, i.e. there is a decision procedure that can do reasoning on the modular ontology in finite time.

Other desiderata for sound semantics that have been considered in the literature include: the ability to cope with inconsistencies [4] and local logic completeness [10]. We believe that the criteria listed above are among the most critical ones for a modular ontology to be semantically sound and practically usable.

The second group of requirements that we consider is aimed at evaluating the language expressivity:

- **Concept Subsumption** (and its special case, **Concept Equivalency**) between modules is probably the most urgently needed feature. For example, the Wine module should be able to extend the Food module with $wine : WineGrape \sqsubseteq food : Grape$.
- **Concept Construction with Foreign Concepts.** It enables a module to build new local concepts based on concepts from other modules, using operators such as negation ($\neg$) , conjunction ($\sqcap$) and disjunction ($\sqcup$).
- **Concept Construction with Role Restrictions.** If R is a role and C is a concept (such that, one or both of them are foreign terms), the language may include existential restrictions ($\exists R.C$), universal restrictions ($\forall R.C$) and qualified number restrictions (e.g., $\leq 2R.C$).
- **Role Inclusion** (and its special case, **Role Equivalency**) between modules. For example, $wine : madeFromGrape \sqsubseteq food : madeFromFruit$.
- **Role Inversion** between modules. For example, $wine : madeIntoWine$ is the inverse of $food : madeFromFruit$.
- **Role Construction**, such as role complement ($\neg R$) , conjunction ($R \sqcap Q$) and disjunction ($R \sqcup Q$), where R and/or Q may be roles from different modules.
- **Transitive Role**, which allows the usage of a foreign transitive role, e.g., the Wine module reuses a transitive role *locatedIn* in the Region module.
- **Nominal Correspondence**. For example, the Wine module declares that the local individual CA is the same as the individual $California$ in the Region module.

Not all applications require the full expressive power of modular ontologies. Based on different intended application scenarios, a specific modular ontology language may only contain a proper subset of the given expressivity features. For example, DDL covers concept subsumption and nominal correspondence, and $\mathcal{E}$-connections addresses only concept and role construction with a special type of roles called “links”.

4 The Abstract Modular Ontology

This section studies an extended type of DFOL, an Abstract Modular Ontology (AMO) language, that will serve as the common testbed for investigating existing approaches according to the criteria introduced above.

4.1 An Abstract Modular Ontology Language

“Ontology is the science of being”(Aristotle, *Metaphysics*). In a general sense, a modular ontology is a set of individual descriptions of the same domain (e.g., Food) that represent correlated, but not identical points of view of multiple observers, or agents. Thus, each ontology module can be seen as describing a point of view held by an agent with respect to the entities (objects) and their relations in the domain. We say that a *domain relation* r_{ij} reflects the ability of an agent j to explain the point of view of an agent i, therefore it is a subjective *belief* rather than an objective *description*.

The idea presented here is strongly influenced by the Local Model Semantics [6] (which has also been influential on DFOL). However, instead of assuming a single relation between each pair of agents as in DFOL, we assume that each agent may need to interact with another agent throught different roles in different contexts. For example, a company can both buy and sell products from and to another company. Consequently, there may be multiple domain relations between ontologies held by a pair of agents.

A DFOL knowledge base (KB) [7] 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 KB. 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. An **Abstract Modular Ontology** (AMO) is a DFOL KB in which each component language L_i is a subset of description logics (DL).

A model of AMO includes a set of local models and domain relations. For each L_i, there exists an interpretation domain Δ_i. Two domains Δ_i and Δ_j are *not necessarily* disjoint. Let M_i be the set of all DL models of L_i on Δ_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 $\Delta_i \times \Delta_j$. A domain relation r_{ij} represents the capability of the module j to map the objects of Δ_i into Δ_j. Each pair of local models may have multiple domain relations, each denoted by r_{ij}^{R} where R is the name for the domain relation. For any domain relation r_{ij}^{R}, we use $\langle d, d' \rangle \in r_{ij}^{R}$ to denote that from the point of view of j, the object d in Δ_i is mapped to the object d' in Δ_j, via relation R. In particular, a special domain relation $r_{ij}^{\rightarrow}$ (read as "image") implies that the object d' in the j's point of view denotes the same entity as the object d in the i's point of view; d' is an image of d and d is a pre-image of d'. Note that the image relations, in general, are *not necessarily* one-to-one. Finally, $r_{ij}^{R}(d)$ denotes the set $\{d' \in \Delta_j | \langle d, d' \rangle \in r_{ij}^{R}\}$. For a subset $D \subseteq \Delta_i$, $r_{ij}^{R}(D)$ denotes $\cup_{d \in D} r_{ij}^{R}(d)$.

Example 1. *An ontology contains two modules $L_{\{1,2\}}$. L_1 contains knowledge about food objects and their relations, such as Grape $\sqsubseteq$ Fruit (a 1-formula). L_2 contains knowledge about wine objects and their relations, such as $Wine \sqsubseteq \forall madeFrom.Grape$. The local domain Δ_1 has Grape objects* ThompsonSeedless, CabernetFrancGrape, *and local domain Δ_2 has Wine object* KathrynKennedy-Lateral *and Grape object* WineGrape. *The image domain relation $r_{21}^{\rightarrow}$ is $\langle 2 : \textsf{WineGrape}, 1 : \textsf{CabernetFrancGrape} \rangle$, while the image domain relation $r_{12}^{\rightarrow}$ is $\langle 1 : \textsf{ThompsonSeedless}, 2 : \textsf{WineGrape} \rangle, \langle 1 : \textsf{CabernetFrancGrape}, 2 : \textsf{WineGrape} \rangle$, and another domain relation $r_{12}^{madeWine}$ is $\langle 1 : \textsf{CabernetFrancGrape}, 2 : \textsf{KathrynKennedyLateral} \rangle$. $r_{12}^{\rightarrow}(1 : \textsf{ThompsonSeedless}) = \{2 : \textsf{WineGrape}\}$. Note* *that* $r_{\overrightarrow{21}} \neq (r_{\overrightarrow{12}})^{-}$ *since* L_1 *does not regard* $1 : \mathsf{ThompsonSeedless}$ *as an image of* $2 : \mathsf{WineGrape}$.

4.2 Expressivity of Abstract Modular Ontology

Consider an agent j is observing the point of view of agent i and finds i uses x to identify an entity (e.g. a grape) in the world, which is identified by j as y. Therefore, j creates an image domain relation $i : x \rightarrow j : y$. If j finds that a set of objects in i's point of view is grouped as $Grape^{m_i}$ by i, then j will regard $r_{\overrightarrow{ij}}(Grape^{m_i})$ as "these objects in my point of view correspond to the *concept Grape* from agent i's point of view". Thus, j can also map *relations* in i's mind to its local point of view: for any relation instance $\langle x_1, x_2 \rangle$ in $\Delta_i \times \Delta_i$, j will regard $r_{\overrightarrow{ij}}(x_1) \times r_{\overrightarrow{ij}}(x_2)$ as a proper image of the relation. It should also be kept in mind that r_{ij} is always a relation viewed from j's point of view. For example, the fact that a person x thinks "y is my best friend" doesn't necessarily mean that y thinks "I'm x's best friend".

Therefore, the image of an i-concept C or i-role P in j is:

- $C^{i \rightarrow j}$: $r_{\overrightarrow{ij}}(C^{m_i})$
- $P^{i \rightarrow j}$: $\bigcup_{\langle x,y \rangle \in P^{m_i}} r_{\overrightarrow{ij}}(x) \times r_{\overrightarrow{ij}}(y)$

Similarly, pre-image of a j-concept D or a j-role R in i is defined as

- $D^{i \leftarrow j}$: $(r_{\overrightarrow{ij}})^{-}(D^{m_j})$
- $R^{i \leftarrow j}$: $\bigcup_{\langle x,y \rangle \in R^{m_j}} r_{\overrightarrow{ij}}^{-}(x) \times r_{\overrightarrow{ij}}^{-}(y)$

A concrete modular ontology language, such as DDL, $\mathcal{E}$-Connections or P-DL, usually contains a set of semantic relation rules, e.g. bridge rules (DDL), $\mathcal{E}$-connection, or concept importings (P-DL), between two ontology modules. Serafini et al. [16] have noted that such rules can be mapped to DFOL interpretation constraints in the form of $i : \phi(x_1, ..., x_n) \rightarrow j : \psi(y_1, ..., y_n)$, where ϕ, ψ are n-ary predicates and $\langle x_i, y_i \rangle$ is connected by a domain relation r_{ij}. Note that DL concepts are unary FOL predicates and DL roles are binary predicates. Consequently, a semantic relation in AMO will be either a concept inclusion axiom or a role inclusion axiom.

In a more general setting, a modular ontology language may also create third party constraints. For example, module j may reuse i-concept $RedWine$ and k-concept $Beverage$, and locally declare $i : RedWine \sqsubseteq k : Beverage$. However, such a third party constraint can be avoid by an "alias" syntax sugar such that $RedWine^{i \rightarrow j}$ and $Beverage^{k \rightarrow j}$ are given local alias $RedWine'$, $Beverage'$ thus transforming the concept inclusion to the one that connects only j-concepts.

A local concept can also be a complex concept constructed with a foreign role and/or a foreign concept, such as universal restriction (e.g.$\forall R.C$) or existential restriction (e.g.$\exists R.C$), as shown in the Table 1. However, arbitrary combination of the *possible* expressivity features in AMO may even lead to undecidability, since the union of multiple decidable logics may be undecidable[1]. The design of a practical modular ontology language has to be a tradeoff between the expressivity and reasoning complexity.

Table 1. Possible AMO Expressivity Features

	Syntax	**Semantics**
Concept Subsumption	$C \sqsubseteq D$	$C^{i\to j} \subseteq D^{m_j}$
	$C \sqsupseteq D$	$C^{i\to j} \supseteq D^{m_j}$
Concept Negation	$\neg C$	$\overrightarrow{r_{ij}}(\Delta_i \backslash C^{m_i})$
Concept Conjunction	$C \sqcap D$	$C^{i\to j} \cap D^{m_j}$
Concept Disjunction	$C \sqcup D$	$C^{i\to j} \cup D^{m_j}$
Universal Restriction	$\forall R.C$	$\{x \in \Delta_j \vert \forall y \in \Delta_i, (y,x) \in r_{ij}^{R} \to y \in C^{m_i}\}$
	$\forall P.D$	$\{x \in \Delta_j \vert \forall y \in \Delta_j, (x,y) \in P^{i\to j} \to y \in D^{m_j}\}$
	$\forall P.E$	$\{x \in \Delta_j \vert \forall y \in \Delta_k, \exists y' \in \overrightarrow{r_{kj}}(y) \wedge (x,y') \in P^{i\to j} \to y \in E^{m_k}\}$
Existential Restriction	$\exists R.C$	$\{x \in \Delta_j \vert \exists y \in \Delta_i, (y,x) \in r_{ij}^{R}, y \in C^{m_i}\}$
	$\exists P.D$	$\{x \in \Delta_j \vert \exists y \in \Delta_j, (x,y) \in P^{i\to j}, y \in D^{m_j}\}$
	$\exists P.E$	$\{x \in \Delta_j \vert \exists y \in \Delta_k, \exists y' \in \overrightarrow{r_{kj}}(y) \wedge (x,y') \in P^{i\to j}, y \in E^{m_k}\}$
Number Restriction[1]	$\leq nR.C$	$\{x \in \Delta_j \vert \#(\{y \in \Delta_i \vert (y,x) \in r_{ij}^{R}, y \in C^{m_i}\}) \leq n\}$
	$\leq nP.D$	$\{x \in \Delta_j \vert \#(\{y \in \Delta_j \vert (x,y) \in P^{i\to j}, y \in D^{m_j}\}) \leq n\}$
	$\leq nP.E$	$\{x \in \Delta_j \vert \#(\{y \in \Delta_k \vert \exists y' \in \overrightarrow{r_{kj}}(y) \wedge (x,y') \in P^{i\to j}, y \in E^{m_k}\}) \leq n\}$
Role Inclusion	$P \sqsubseteq R$	$P^{i\to j} \subseteq R^{m_j}$
	$P \sqsupseteq R$	$P^{i\to j} \supseteq R^{m_j}$
Role Inverse	P^-	$\{(y,x) \vert (x,y) \in P^{i\to j}\}$
Role Complement	$\neg P$	$(\Delta_j \times \Delta_j) \backslash P^{i\to j}$
Role Conjunction	$P \sqcap R$	$P^{i\to j} \cap R^{m_j}$
Role Disjunction	$P \sqcup R$	$P^{i\to j} \cup R^{m_j}$
Transitive Role	$trans(P)$	$(P^{i\to j})^+ = P^{i\to j}$
Nominal	$\{x\} \to \{y\}$	$y \in \overrightarrow{r_{ij}}(x)$

[1] $\geq$ case is similar.
C is an i-concept, D is a j-concept, E is a k-concept; P is an i-role, R is a j-role, Q is a k-role; x is a i-individual, y is a j-individual; $i \neq j$, $j \neq k$, i may be or may not be k. All formulas represent module j's point of view and constructed concepts (roles) are j-terms. Local domains of modules may be partially overlapping.

4.3 Semantic Soundness of the Abstract Modular Ontology

To precisely specify the semantic soundness of AMO, we need to answer several questions. First, what are the logical consequences in an AMO? How can local constraints in the agents' local points of view influence each other? For example, if agent i thinks "a is b's best friend", and agent j thinks $i : a$ is x and $i : b$ is y in j's mind, will j also hold the constraint that "x is y's best friend"?

Second, if there are inconsistencies in the points of view of two agents, what is the possible cause of such consistencies? For example, if agent j holds the belief that "x is y's enemy", possible causes can be either i and j hold incompatible points of view while the domain relations ($a \to x, b \to y$) are sound, or i and j actually hold compatible points of view but the domain relations are wrong (e.g. j has mistaken z as y and label both y and z as b locally, while z is x's enemy). While the first type of inconsistency is hard to eliminate (subjectivity), are there principled ways to avoid the second type of inconsistency (miscommunication)?

Third, if beliefs of agents are compatible, what is an "objective" way to integrate their knowledge? Or in other words, to "restore" a description of the physical world that reflects the consensus among the agents, such that logical consequences are consistent in the *integrated point of view* and each local point of view. For example, if a person *Alice* (identified as $i : a$ and $j : x$) behaves as the best friend of another person *Bob* (identified as $i : b$ and $j : y$), how can we construct an "integrated" description that is acceptable by both i and j, such that if $i(j)$ asserts a conclusion (e.g. x is y's best friend), the "integrated" description can also confirm the conclusion?

Addressing such problems is critical in identifying and solving several semantic difficulties that arise in modular ontology languages. Next, we introduce some definitions that are useful in precisely stating problems such as those we informally outlined above.

Definition 1 (AMO Satisfiability). *Let $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ be a model for an AMO $O = \{L_i\}$ with interpretation constraint sets $\{C_{ij}\}$ (as defined in Table 1) , where $m_i = \langle\Delta_i, (.)_i\rangle$ is the local interpretation of i (Δ_i is the local domain of i, $(.)_i$ is the assignment function of i) and r_{ij} denotes all domain relations between m_i and m_j, including "image ($\rightarrow$)". We say that M satisfies O, denoted as $M \vDash O$, iff $m_i \vDash L_i$, for all i, and $M \vDash C_{ij}$, for all i and j.*

Definition 2 (AMO Entailment). *An AMO $O = \{L_i\}$ entails $C \sqsubseteq D$, where C, D are j-concepts, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, $m_j \vDash C \sqsubseteq D$.*

Although the above definition only addresses intra-module subsumption, it can be easily extended to inter-module subsumption with a simple syntax rewriting. If C is an i-concept, we can always create a j-concept C' interpreted as $C^{i\rightarrow j}$, and then $i : C \sqsubseteq j : D$ can be transformed as $j : C' \sqsubseteq j : D$.

Definition 3 (Localized and Globalized Semantics). *An AMO $O = \{L_i\}$ has only globalized semantics, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, $M \vDash O$, $m_i = \langle\Delta_i, (.)_i\rangle$, local domains $\{\Delta_i\}$ of $\{L_i\}$ must be identical. Otherwise, it has localized semantics.*

Definition 4 (Decidability). *An AMO $O = \{L_i\}$ is decidable if for every satisfiability problem (therefore also entailment problem) C for i, there exists an algorithm that is capable of deciding in a finite number of steps whether there exists a model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$, $M \vDash O$, such that C is satisfiable in m_i.*

Definition 5 (Directional Semantic Relations). *Domain relations in an AMO are directional, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, for any $i \neq j$, $m_j \vDash C \sqsubseteq D$, doesn't imply that $C^{i\leftarrow j} \subseteq D^{i\leftarrow j}$ must be true in m_i.*

Transitive reusability means that an agent can infer local constraints based on observing constraints in other agents' points of view. For example, if i believes "a is b's best friend", and j believes domain relation $i : a \rightarrow j : x, i : b \rightarrow j : y$, then j may reuse i's knowledge and infer that "x is y's best friend". Furthermore, if another agent k who is confident in j's judgement, and believes $j : x \rightarrow k : p, j : y \rightarrow k : q$, then k also believes "p is q's best friend".

Definition 6 (Transitive Reusability). *For an AMO $O = \{L_i\}$, L_i is said to be* reusable *by j ($j \neq i$) if for any concepts C, D in L_i, such that $L_i \vDash C \sqsubseteq D$,* *we have that for $M = \langle \{m_i\}, \{r_{ij}\} \rangle$ of O, $C^{i \to j} \subseteq D^{i \to j}$ must be true in m_j. L_i is said to be* transitively reusable *if for any j, k $(i \neq j \neq k)$, if L_i is reusable by L_j, and L_j is reusable by L_k, then we must have L_i is reusable by L_k.*

Exact reasoning means that the points of view of all agents can be reconciled into a point of view (a consensus) that is consistent with each individual agent's point of view. Since such a merged state will be the consensus of individual agents, their compatible beliefs may be combined. For example, if i believes x is the identifier of a person *Alice*, j believes a is the identifier of *Alice* and $i : x \to j : a$, then the merged state of the two agents will "believe" $i : x$ and $j : a$ are all identifiers of *Alice*. Thus, all semantic relation rules, in their DFOL form $i : \phi(x_1, ..., x_n) \to j : \psi(y_1, ..., y_n)$ (n=1 or 2), where x_i, y_i are connected by $r_{\overrightarrow{ij}}$, will be reduced to $\phi(x_1, ..., x_n) \to \psi(x_1, ..., x_n)$.

Definition 7 (Exact Reasoning). *Reasoning in an AMO $O = \{L_i\}$ is exact, iff for any model $M = \langle \{m_i\}, \{r_{ij}\} \rangle$ of O, there exists a classical model $M' = \Re(M) = \langle \Delta_m, (.)_m \rangle$, such that $M \vDash \phi \Rightarrow M' \vDash \phi$. $\Re$ denotes the reduction from M to M' as follows:*

- $\Delta_m = \cup_i \Delta_i$
- *The assignment function $(.)_m$ is defined as: for any concept $i : C$, $C^m = C^{m_i}$; for any role $i : P$, $P^m = P^{m_i}$; for any individual $i : I$, $I^m = I^{m_i}$.*
- *for every image domain relation, if $(i : x, j : y) \in r_{\overrightarrow{ij}}$, add $i : x = j : y$.*
- *for every other domain relation R, if $(i : x, j : y) \in r^R_{ij}$, assign (x, y) to R^m.*

5 Semantic Soundness of Existing Approaches

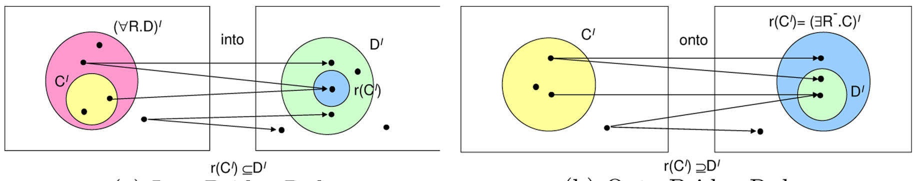
5.1 Distributed Description Logics

Distributed Description Logics (DDL) [4], adopts a "linking"-based approach. In DDL, the semantic mappings between disjoint modules L_i and L_j are established by a set of "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: $r_{ij}(\phi^{m_i}) \supseteq \psi^{m_j}$
- Individual Correspondence: $i : a \to j : b$, semantics: $b^{m_j} \in r_{ij}(a^{m_i})$

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}, and can be seen as the image domain relation $r_{\overrightarrow{ij}}$ in AMO. Although ϕ, ψ may be role names[5,16], semantics and decidability of such an extension is still not well-understood. The semantics of bridge rules between concepts is shown in Figure 1.

Distributed concept correspondence between two modules in DDL covers some of the most important scenarios that require mapping between ontology modules. Since DDL has clear DFOL interpretation, it is easy to see that it has localized semantics and supports directional semantic relations. DDL is decidable if each connected module is decidable [4,12].



(a) Into Bridge Rules (b) Onto Bridge Rules

Fig. 1. Semantics of DDL Bridge Rules between Concepts

However, DDL, as noted in [9,8], doesn't ensure transitive reusability and exact reasoning: (a) *Subsumption Propagation problem*: concept subsumption links in DDLs do not propagate transitively. For example, in the case of 3 ontology modules $L_{\{1,2,3\}}$, the bridge rules $1 : Bird \xrightarrow{\sqsupseteq} 2 : Fowl$ and $2 : Fowl \xrightarrow{\sqsupseteq} 3 : Chicken$ do not in general ensure that $1 : Bird \xrightarrow{\sqsupseteq} 3 : Chicken$; (b) *Inter-module Unsatisfiability problem*: DDLs may not detect unsatisfiability across ontology modules. For example, $1 : Bird \xrightarrow{\sqsupseteq} 2 : Penguin$ and $1 : \neg Fly \xrightarrow{\sqsupseteq} 2 : Penguin$ do not render $2 : Penguin$ unsatisfiable even if L_1 entails $Bird \sqsubseteq Fly$.

A primary source of such difficulties has to do with the fact that the domain relations in DDL can be arbitrary [2]. In the absence of a formal mechanism to prevent inconsistency between the agents' points of view due to miscommunication, domain relations cannot be reused by other modules (r_{13} cannot be inferred from r_{12} and r_{23}, as illustrated by example (a) above). This precludes transitive reusability. Furthermore, unsatisfiability across ontology modules can not be detected (as illustrated by example (b) above) since objects of disjoint i-concepts can be mapped to the same object in j. Bao et.al. [2] have recently shown that one-to-one domain relation is a sufficient condition for exact DDL reasoning. However, at present, there is no principled approach to coming up with such domain relations in DDL alone.

5.2 $\mathcal{E}$-Connections

While DDL allows only one type of domain relations, the $\mathcal{E}$-connection approach allows multiple "link" relations between two domains. $\mathcal{E}$-connections between DLs [11,9] restrict the local domains of the $\mathcal{E}$-connected ontology modules to be disjoint (therefore ensure localized semantics). 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 $E \in \mathcal{E}_{ij}$, 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 a $\mathcal{E}$-connection E.

An advantage of $\mathcal{E}$-connections is that a collection of $\mathcal{E}$-connected ontology modules is decidable if all modules are decidable [12]. However, since there are no image domain relations in $\mathcal{E}$-connections, transitive reusability cannot be guaranteed in general. Some scenarios may still allow knowledge propagation. For example, if module i contains $D \sqsubseteq E$, module j contains $A \equiv \forall R.D$, $B \equiv \forall R.E$, where R is a $\mathcal{E}$-connection from j to i, j can infer that $A \sqsubseteq B$ must be true. $\mathcal{E}$-connections are also directional.

The exactness of reasoning (as defined in Definition 7) of $\mathcal{E}$-connections given in [9] can be guaranteed since there is no image domain relation. A reduction from a $\mathcal{E}$-connections model to a classical model can be obtained by constructing a simple union of all local models where all link instances are converted into classic role instances. However such a reduction does not hold in the case of "generalized links" [13] where a link/role name can be used within different contexts. For example, given two modules $L_{\{1,2\}}$, L_2 contains $Penguin \sqsubseteq \forall isa^{(1)}.(1 : Bird)$ and $Penguin \sqsubseteq \exists isa^{(2)}.(2 : PolarAnimal)$, where isa is interpreted as link or local role under different contexts. Since $1 : Bird$ and $2 : PolarAnimal$ are disjoint by default, the disjoint union of isa interpretation in each of its contexts will be unsatisfiable.

5.3 Package-Based Description Logics

Package-based Description Logics (P-DL)[3] offer a tradeoff between the strong module disjointness assumption of DDL and $\mathcal{E}$-connections, and on the other hand, the OWL importing mechanics, which forces *complete overlapping* of modules. 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}. In what follows, we will examine a restricted type of package extension which only allows import of concept names.

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

- Every importing relation is one-to-one in that it maps each object of t^{m_i} to a single unique object in t^{m_j}, therefore $(r_{\overrightarrow{ij}})^t(t^{m_i}) = t^{m_j}$.
- Term Consistency: importing relations of different terms are consistent, i.e., for any $i : t_1 \neq i : t_2$ and any $x, x_1, x_2 \in \Delta_i$, $(r_{\overrightarrow{ij}})^{t_1}(x) = (r_{\overrightarrow{ij}})^{t_2}(x)$ and $(r_{\overrightarrow{ij}})^{t_1}(x_1) = (r_{\overrightarrow{ij}})^{t_2}(x_2) \neq \emptyset \rightarrow x_1 = x_2$.
- Compositional Consistency: if $(r_{\overrightarrow{ik}})^{i:t_1}(x) = y_1$, $(r_{\overrightarrow{ij}})^{i:t_2}(x) = y_2$,$(r_{\overrightarrow{jk}})^{j:t_3}$ $(y_2) = y_3$, , (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 *image domain relation* between m_i and m_j is $r_{\overrightarrow{ij}} = \cup_t (r_{\overrightarrow{ij}})^t$ and is strictly one-to-one. From the multi-agent point of view, such a domain relation ensures unambiguous communication between each modules. Consequently, r_{ij} in a P-DL model isomorphically "copies" the relevant partial domain from m_i to m_j (Figure 2). Since the construction of a local model is dependent on the structure of local models of imported modules, P-DL allows a relaxation of the domain disjointedness assumption adopted in DDL and $\mathcal{E}$-connections. However, the loss of disjointedness does not sacrifice *localized semantics* property of modules, since they are, unlike in OWL, only partially overlapping. Consequently, there is no required global model.

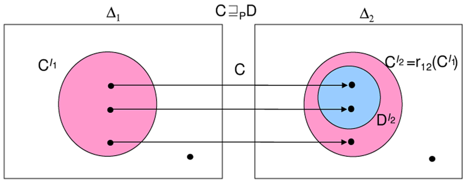


Fig. 2. Semantics of P-DL

With a principled way to avoid semantic imprecision, P-DL can ensure transitive reusability and exact reasoning [2]. A reduction from a P-DL model to a classical DL model is the union of all local models with"copied" objects being merged. However, semantic relations in P-DL may not always be directional in local domains that overlap: if module j imports concept C, D from i, then $C \sqsubseteq D$ in j will imply $C^{i \leftarrow j} \sqsubseteq D^{i \leftarrow j}$ in Δ_i.

The general decidability transfer property does not always hold in P-DL since the union of two decidable fragments of DL may be undecidable [1]. This presents semantic difficulties in the general setting of connecting ADSs [12]. However, in a setting where different ontology modules are specified using subsets of the *same* decidable DL language, such as $\mathcal{SHOIQ}(D)$ (OWL-DL), and importing is only allowed for concept names, the union of such modules is decidable. In such a setting, semantics-preserving reduction from P-DL model to the integrated DL model is available [2] making P-DL decidable.

The comparison is summarized in Table 2.

Table 2. Comparison of Semantic Soundness

	Localized Semantics	**Exact Reasoning**	**Directional Relation**	**Transitive Reusability**	**Decidability***
DDL	Yes	No	Yes	No	Yes[1]
$\mathcal{E}$-Connections	Yes	Partial[2]	Yes	Partial	Yes
OWL	No	Yes	No	Yes	Yes (OWL-DL)
P-DL	Yes	Yes	Partial	Yes	Partial[3]

* when each local module is decidable; [1] yes only for concept bridge rules; [2] yes without generalized links; [3] yes for concept importing and each module from a subset of $\mathcal{SHOIQ}(D)$.

Table 3. Comparison of Expressivity of DDL, $\mathcal{E}$-connections and p-DL

	Syntax	**DDL**	**$\mathcal{E}$-Connections**	**P-DL**
Concept Subsumption	$C \sqsubseteq D$	$\surd$	$\times$	$\surd$
	$C \sqsupseteq D$	$\surd$	$\times$	$\surd$
Concept Negation	$\neg C$	$\surd$	$\times$	$\surd$
Concept Conjunction	$C \sqcap D$	$\surd$	$\times$	$\surd$
Concept Disjunction	$C \sqcup D$	$\surd$	$\times$	$\surd$
Universal Restriction	$\forall R.C$	$\times$	$\surd$	$\surd$
	$\forall P.D$	$\times$	$\times$	$\times$
	$\forall P.E$	$\times$	$\times$	$\times$
Existential Restriction	$\exists R.C$	$\times$	$\surd$	$\surd$
	$\exists P.D$	$\times$	$\times$	$\times$
	$\exists P.E$	$\times$	$\times$	$\times$
Number Restriction[1]	$\leq nR.C$	$\times$	$\surd$	$\surd$
	$\leq nP.D$	$\times$	$\times$	$\times$
	$\leq nP.E$	$\times$	$\times$	$\times$
Role Inclusion	$P \sqsubseteq R$	$\surd$	$\times$	$\times$
	$P \sqsupseteq R$	$\surd$	$\times$	$\times$
Role Inverse	P^-	$\times$	$\times$	$\times$
Role Complement	$\neg P$	$\times$	$\times$	$\times$
Role Conjunction	$P \sqcap R$	$\times$	$\times$	$\times$
Role Disjunction	$P \sqcup R$	$\times$	$\times$	$\times$
Transitive Role	$trans(P)$	$\times$	$\surd$[2]	$\times$
Nominal	$\{x\} \to \{y\}$	$\surd$	$\times$	$\times$

[1] $\geq$ case is similar. [2] only with generalized links.
Notations are as the same in Table 1. All formulas represent module j's point of view.

6 Expressivity of Existing Approaches

Table 3 shows the expressivity comparison of DDL, $\mathcal{E}$-connections and P-DL.

6.1 Distributed Description Logics

DDL bridge rules cannot be directly read as inter-module concept subsumptions. However, several techniques have been studied to simulate concept subsumptions with bridge rules [17]. DDL is also capable of using foreign concepts in local concept negation, conjunction and disjunction by a simple "alias" syntax sugar as mentioned in section 4.

However, major limitations in the expressivity of DDL have to do with linking modules with roles. Since DDL semantics only allows one type of domain relations, role instances cannot be created between local models. This precludes concepts built with foreign role or inter-module role construction. Although the extended DDL in [5,16] allows role inclusions, the decidability of such an extension as well reasoning algorithms that work in such a setting are still unknown.

6.2 $\mathcal{E}$-Connections

In contrast with DDL, $\mathcal{E}$-connections allow role connections between modules but doesn't allow inter-module concept inclusion. Although it has been argued that $\mathcal{E}$-connections are more expressive than DDLs [12,8], the intended use of DDL bridge rules and $\mathcal{E}$-connection links are quite different. This is made clear in the AMO framework, where DDL bridge rules are interpreted as image domain relations, and $\mathcal{E}$-connection links are not image domain relations. Since $\mathcal{E}$-connections strictly require local domain disjointedness, no direct inter-module concept subsumption can be allowed. Therefore, DDL bridge rules and $\mathcal{E}$-connection links actually cover different application scenarios, and thus are complementary in their roles.

It should be noted that the direction of "links" in $\mathcal{E}$-connections is the inverse of AMO roles defined in Table 1 and 3. In $\mathcal{E}$-connections, module i can use a link from i to j to construct an i-concept, whereas in AMO, i can only construct i-concepts with a role from j to i. This difference arises from AMO's underlying assumption that any domain relation r_{ij}^{R} is only a subjective point of view of j and should only be used in j. Therefore, we contend that a $\mathcal{E}$-connections link used in module i, although is syntactically given as from i to j, stands for the subjective point of view of i, not j. The difference can be syntactically eliminated by inverting E-connection links to obtain AMO roles.

However, the expressivity of $\mathcal{E}$-connections is limited by the need to ensure the disjointedness of local domains. Thus, a concept cannot be declared as a subclass of a foreign concept, and foreign concepts cannot be used in local concept constructions. A property cannot be declared as sub-relation of a foreign property, and neither foreign classes nor foreign properties can be instantiated. It is also difficult to combine $\mathcal{E}$-connections and OWL importing [8].

$\mathcal{E}$-connection links cannot be seen as foreign roles in AMO. Their usage is equivalent to allowing an AMO local role to have foreign concepts within its range. In the case of the generalized link property [13], the boundary between local roles and links is further ambiguous. The $\mathcal{E}$-connections syntax proposal [8] requires that the source of a link be the module in which it has been declared. Therefore, link constructors, such as inverse, conjunction, disjunction and complement, are different from the inter-module role constructors defined in Table 1, and are closer to intra-module role constructors.

6.3 Package-Based Description Logics

P-DL expressivity features summarized in table 3 only allow concept name importing, since decidability of more expressive variants of P-DL is still unknown. Despite its stronger domain relation restrictions (ont-to-one), P-DL is more expressive than DDL and $\mathcal{E}$-Connections in several ways. P-DL allows inter-module concept subsumption, concept construction with foreign concepts, and connecting modules with roles, thus provides several expressivity features that are missing either in DDL or $\mathcal{E}$-connections.

DDL with only concept correspondence and $\mathcal{E}$-connections without generalized links can be reduced to P-DL. For example, an into rule $i : C \xrightarrow{\sqsubseteq} j : D$ in DDL can be reduced to a P-DL axiom $C \sqsubseteq D$ in module j and C is an imported concept; A $\mathcal{E}$-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 given Table 1; $\forall(i : E).(j : D)$ can be constructed in a similar fashion.

Therefore, we believe that the importing approach adopted by P-DL which relaxes the strong module disjointedness assumption of DDL offers the possibility of avoiding many of the semantic difficulties of current modular ontology language proposals while improving the expressivity.

7 Conclusions

This paper provides a formal investigation of the motivation, evaluation criteria, an abstract framework of modular ontologies, and compares the semantic soundness and expressivity of several modular ontology languages. The main contributions of this paper are: a) identification and precise definition of possible requirements for semantically sound modular ontology languages; b) identification of desirable expressivity features of modular ontology languages; c) introduction of an Abstract Modular Ontology (AMO) framework which offers a basis for comparing different modular ontology languages; d) comparison of the semantic soundness and the expressivity of DDL, $\mathcal{E}$-connections and P-DL; and e) analysis of several semantic difficulties and expressivity limitations of DDL and $\mathcal{E}$-connections, and propose an approach in the form of a partial importing mechanism in P-DL to overcome such limitations.

We conclude that different existing modular ontology language proposals are motivated by, and hence are responsive to, different application scenarios. At present, there is no modular ontology language with known decidability and inference complexity that supports both general inter-module concept and inter-module role correspondence and satisfies all semantic soundness requirement. Our results suggest that in order to improve the expressivity of existing modular ontology languages, and to ensure their semantic soundness, the strict module disjointedness assumption adopted by DDL and $\mathcal{E}$-connections may need to be at least partially relaxed. Work in progress is aimed at the development of a reasoning algorithm for an expressive and semantically sound modular ontology language, e.g. P-DL.

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. On the semantics of linking and importing in modular ontologies (extended version). Technical report, TR-408 Computer Sicence, Iowa State University, 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/DOA/ODBASE*, 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.
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. Working with multiple ontologies on the semantic web. In *International Semantic Web Conference*, pages 620–634, 2004.
10. B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *KR2006*, 2006.
11. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of description logics. In *Description Logics Workshop, CEUR-WS Vol 81*, 2003.
12. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.
13. B. Parsia and B. C. Grau. Generalized link properties for expressive epsilon-connections of description logics. In *AAAI*, pages 657–662, 2005.
14. 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.
15. G. Schreiber and M. Dean. Owl web ontology language reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/, February 2004.
16. L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In *IJCAI*, pages 576–581, 2005.
17. L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376, 2005.