Divide and Conquer Semantic Web with Modular Ontologies - A Brief Review of Modular Ontology Language Formalisms

Jie Bao and Vasant G Honavar

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

Abstract. Distributed data and knowledge base applications need ontology languages and tools that can support collaborative construction, sharing, and use of large ontologies. Modular ontology languages aim to address this challenge by providing language support for organizing complex ontologies into relatively independent modules, selective sharing of information among ontology modules, and reasoning with multiple ontology modules. This paper summarizes the evolution, language features and semantics of recent modular ontology language proposals, including Distributed Description Logics (DDL),$\mathcal{E}$-Connections and Package-based Description Logics (P-DL) and describes some steps towards a modular ontology language that meets the needs of real-world Semantic Web applications.

1 Introduction

The rapid growth and adoption of the world-wide web was possible in part because it allowed a large community of individuals around the world to contribute to its construction by linking pages created by individuals or groups via hyperlinks. Similarly, we expect that effective tools that would enable individuals with expertise in specific areas to contribute ontology modules that can be conceptually linked into larger ontologies would significantly accelerate the realization of the vision of the semantic web [10].

A typical large-scale ontology construction scenario is given in the following: The animal genomics community consists of several autonomous, geographically dispersed, research groups around the world. There is an urgent need for an animal trait ontology (ATO) for a diverse set of species (e.g., for cross-species comparisons). No single research group possesses all of the expertise needed to construct the desired ATO. Consequently, it is necessary and natural for groups with different areas of expertise (e.g., species-specific expertise about horses, chicken, pigs, etc.) to work more or less independently to create ontology modules that can then be linked together as needed. Because multiple groups might hold different ontological commitments, terminological clashes or conceptual differences between the groups (and hence the ontology modules created by them) are simply unavoidable. Hence, there is a need for mechanisms for linking ontology modules so as to preserve the semantic locality while ensure a partial consensus on publicly shared knowledge. Furthermore, inherent inefficiencies (with regard to memory and processing time needs) in the use (e.g. editing, reasoning, communicating) of large ontologies can be minimized by taking advantage of the modular nature of the ontologies.

This argues for a modular approach to design and use of complex ontologies wherein ontologies such as ATO, instead of being treated as a monolithic entity, are organized into modules that reflect the organizational structure of knowledge in a domain of interest. Thus, it is natural to organize ATO (at a fairly high level), in terms of species-specific ATO modules (e.g., those that focus on horse, cattle, chicken, etc).

Unfortunately, the current state of the art in ontology engineering is reminiscent of the state of software engineering nearly four decades ago: Today's ontology languages, like the very early programming languages, are largely unstructured, and offer little support for modular design of ontologies, of selective knowledge sharing between ontology modules. As a consequence, many existing ontologies are difficult to reuse in a larger context, leading to an *ontology engineering bottleneck*, which is a significant hurdle in the large-scale design, development, and deployment of semantic web applications.

We illustrate some of the limitations of current ontology languages using the relatively well-known Wine ontology as an example. The Wine ontology, often used to demonstrate the features of OWL, the popular web ontology language, is given in two OWL files [1] connected by `owl:imports`, focused on wine knowledge and general food knowledge, respectively. However, the ontology also presents many problems due to the limitations of OWL and the current, largely undisciplined approach to ontology engineering:

- Lack of support for *localized semantics*. The OWL semantics [31] requires all ontologies that are connected with `owl:imports` only share a global interpretation. Thus, in order for the Wine and the Food ontologies to be used together for reasoning, the two ontologies have to be combined into one ontology. Hence, "modularization" with `owl:imports` offers only a syntactical solution, not a satisfactory semantic solution to reasoning with ontology modules. In many applications, combining the relevant ontologies into an integrated ontology is not possible due to privacy, copyright or security concerns.
- Lack of support for *partial reuse*. For example, the Food ontology contains knowledge about grapes and other foods; however, the Wine ontology has to import *all* of the terms and axioms in the Food ontology although it only needs axioms and terms associated with *grapes*. In general, an OWL ontology no matter how large it is, has to be either completely reused or completely discarded. Because an ontology may have to refer to (and hence import terms and axioms from) some other ontologies, while not only directly imported ontologies, but also indirectly imported ontologies, are forced to be reused in their entirety, defining a relatively small new ontology may involve the use of a large subset of known Semantic Web ontologies.
- Lack of *fine-grained organization*. For example, knowledge about geographic region in wine.rdf is not specific to wine, but instead is general knowledge that may be reused in other contexts. But since it is intertwined with other parts of wine.rdf, it does not lend itself to reuse in other applications.
- Lack of formal support for *collaborative ontology building*. At present, collaboration in ontology construction requires informal commitment among the collaborators; and despite the increasing use of tools like CVS, it requires intensive human

[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

communication, e.g. emails [9]. For example, there is no language support to enable two wine experts from U.S. and France to *concurrently* work on *different parts* of the wine ontology.

Consequently, there is an increasing interest in modular ontology languages. Several proposals have been made including: Distributed Description Logics (DDL) [11], $\mathcal{E}$-Connections [25, 21] and Package-based Description Logics (P-DL) [8]. This paper summarizes the evolution, language features and semantics of recent modular ontology language proposals, and describes some steps towards a modular ontology language that meet the needs of real-world Semantic Web applications.

This paper is not intended to be a complete survey of modular ontologies. Some important aspects of modular ontologies are not covered in this paper include *modularization* of existing ontologies [38, 22, 32, 39], reasoning (as well as its decidability and complexity results) with modular ontologies [21, 20, 36, 7, 33, 35], collaborative building modular ontologies [9, 8], inconsistency handling in modular ontologies [28, 29, 12], as well as related problems such as ontology mapping, matching, alignment and integration. Interested readers may refer to the papers cited above for details.

2 Evolution of Modular Ontology Languages

Some of the modular ontology ideas can be traced to studies of knowledge engineering over the past few decades. The Cyc project [27], started from 1984, divides the huge Cyc knowledge base (expressed in the CycL language) into many *microtheories* - collections of concepts and facts pertaining to particular knowledge domains. Partition-based Logics [1] provides an approach to automatically decompose propositional and first-order logic (FOL) into *partitions* and an algorithm for reasoning with such partitions using message passing. These efforts provided the important initial experiences with building and reasoning with modular knowledge bases.

Both CycL and Partition-based Logics do not provide *localized semantics* for knowledge modules or principled ways of connecting ontology modules, Neither do they support partial reuse or mechanisms for controlling semantic heterogeneity among knowledge base modules. Indeed, even reusing a small portion of Cyc, OpenCyc [2], because of lack of modularity, requires the entire OpenCyc ontology to be loaded although only a small part of it may be of interest. The OWL scaffolding version of OpenCyc v0.7.8b ($>$700MB), containing assertions for over 60,000 Cyc constants, “takes approximately 9 hours to load into Protege” (from OpenCyc homepage, 2004/06/04 announcement).

CycL language, being a variant of FOL, is not in general, decidable. Modern ontology languages like OWL, are based on description logics - decidable fragment of FOL. They are supported by highly efficient reasoners e.g. FaCT [23].

Recent work on modular ontology languages is heavily influenced by contextual logics, and in particular, the **Local Models Semantics** (LMS) [18]. LMS theory allows a family of logic languages to have *local models* that represent the local semantic points of view of each of the languages. A formula in one language may be the logical consequence of a formula in another language. Thus, LMS provides a practical tradeoff between locality and compatibility in multi-context knowledge bases.

[2] http://www.opencyc.org/

A **Distributed First Order Logics** (DFOL) knowledge base (KB) [16] (and hence, a DFOL ontology) includes a family of first order languages, each represents a piece of the global knowledge. DFOL semantics includes a set of *local models* (first order interpretations) for each of the language, and a set of *domain relations* between objects in those local models. Inference with DFOL is enabled by a sound and complete calculus as an extension of natural deduction that allows theorem exporting (a theorem can be the logical consequence of another theorem in a different language) among different languages.

Based on DFOL, **Distributed Description Logics** (DDL) [11] allows directional relations among multiple description logics. The initial proposal provides *bridge rules* between concepts and individuals in different ontologies in the form of:

$i : C \xrightarrow{\sqsubseteq} j : D$ (INTO)

$i : C \xrightarrow{\sqsupseteq} j : D$ (ONTO)

$i : x \mapsto j : y$ (partial individual correspondence)

$i : x \overset{=}{\mapsto} j : \{y_1, y_2, ...\}$ (complete individual correspondence)

where C, D are concepts, x, y are individuals, i, j indicate indexes of ontologies. INTO and ONTO rules are meant to simulate concept subsumptions across ontologies. For example, there may be bridge rules as

$i : Dog \xrightarrow{\sqsubseteq} j : Pet$

$i : Animal \xrightarrow{\sqsupseteq} j : Pet$

Individual correspondences in DDL allow one-to-one or one-to-many mappings between individuals across ontologies, such as

$i : US \mapsto j : United\_States$

$i : NYC \overset{=}{\mapsto} j : \{Bronx, Manhattan, Queens, Brooklyn, StatenIsland\}$

The first syntax proposal influenced by the DDL notion is **CTXML**(ConTeXt Markup Language) [13], an ontology mapping language across XML-based hierarchies. It allows mapping relations $\xrightarrow{\sqsupseteq}$(onto, or more general than), $\xrightarrow{\sqsubseteq}$(into, or less general than), $\xrightarrow{=}$(equivalent), $\xrightarrow{*}$(compatible) or $\xrightarrow{\perp}$(disjoint) between concepts on different hierarchies. These notions have been incorporated into OWL to obtain the **C-OWL** language [14]. C-OWL provides a syntax for DDL which allows specification of the five types of mappings between concepts described above, between roles or between individuals in different OWL ontologies.

Subsequent extensions to DDL include heterogenous mapping between roles and concepts [17]. For example, `marriage` relation can be represented by a concept $Marraige$ in one ontology but by a role $marriesTo$ in other ontology; a concept/role bridge rule can be declared as

$Marriage \xrightarrow{\sqsubseteq} marriesTo$ and

$Marriage \xrightarrow{\sqsupseteq} marriesTo$

to indicate $Marriage$ instances are always linked to certain pairs of individuals (as $marriesTo$ instances) of the other ontology.

However, DDL has significant limitations with regard to linking of modules with roles. For example, roles defined in other ontology modules (i.e. foreign roles) cannot be used to construct new concepts, or to construct new roles from foreign roles.

On the contrast, $\mathcal{E}$**-Connections** [26] focus on offering inter-module role connections. Some of the ideas incorporated into $\mathcal{E}$-connections can be traced back to the fusion of abstract description systems (ADS) [3], 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 DLs [24, 21] 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). For example, two concepts $i : PerOwner$ and $j : Pet$ can be connected with a link $owns$ such that:

$i : PerOwner \sqsubseteq \exists owns.(j : Pet)$

Such a division of links and local roles ensures the decidability transfer property: if all ontologies connected by E-connections (the set of links) are locally decidable, then their union is also decidable [24].

An XML Syntax of E-Connections is first provided in [21] for the e-connected version of OWL-Lite, denoted as $C^{\mathcal{E}}(\mathcal{SHIF}(D))$. More expressive extensions are reported in [19] as $C^{\mathcal{E}}_{\mathcal{IHN}^+{}_s}(\mathcal{SHOIN}(D))$ which allow each connected modules to be a subset of OWL-DL (i.e. $\mathcal{SHOIN}(D)$)) which includes $\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO}$. Subscripts $_{\mathcal{IHN}^+{}_s}$ stand for several role relations of the form:

$_{\mathcal{I}}$: $owns = ownedBy^-$ (link inverse)

$_{\mathcal{H}}$: $sonOf \sqsubseteq childOf$ (link inclusion)

$_{\mathcal{N}}$: $PetMania \sqsubseteq (\geq 5\ owns.\top_j)$ (link number restriction)

$^+$: $Trans(largerThan)$ (transitive link)

$()_s$: $Symmetric(brotherOf)$ (symmetric link)

An extension of transitive link, called generalized link is also reported in [30], which can control transitivity of links among modules.

$\mathcal{E}$-connections do not allow inter-module concept inclusion as DDL designed to express. Since $\mathcal{E}$-connections strictly require local domain disjointedness, no direct inter-module concept subsumption can be allowed. Furthermore, a concept cannot be declared as a subclass of a foreign concept, and foreign concepts cannot be used in local concept constructions. A role cannot be declared as sub-role of a foreign role. Neither foreign classes nor foreign roles can be instantiated. Neither links and local roles, nor links that connect different pairs of ontologies (e.g. links E_{ij} and E_{jk}), can be used together to construct new roles or links. It is also difficult to combine $\mathcal{E}$-connections and OWL importing [19].

Although it has been argued that $\mathcal{E}$-connections are more expressive than DDLs [25, 19], DDL and $\mathcal{E}$-connections actually cover different application scenarios, and thus are complementary in their expressivity. On the other hand, they both conform to the "linking" approach such that term sets (or called signatures) of ontology modules are disjoint, and semantic relations between modules is only given by mappings like DDL bridge rules and $\mathcal{E}$-connection links. This restriction limits expressivity and presents inference difficulties [6, 5].

Package-based Description Logics (PDL) [8] provides an alternative solution, i.e., a selective importing approach, to improve the expressivity and inference soundness for modular ontologies. In a P-DL ontology, the whole ontology is composed of a set of packages. Terms (such as $Dog, Animal$) and axioms (such as $Dog \sqsubseteq Animal$) are defined in specific home packages. A package can directly use terms defined in another package. In other words, an existing package can be *imported* into another package. For example, an ontology O has two packages:

$$\mathbf{P_{Animal}}$$

(1a) $1 : Dog \sqsubseteq 1 : Carnivore$
(1b) $1 : Cat \sqsubseteq 1 : Carnivore$
(1c) $1 : Carnivore \sqsubseteq \forall 1 : eats.(1 : Animal)$

$$\mathbf{P_{Pet}}$$

(2a) $2 : PetDog \sqsubseteq 1 : Dog$
(2b) $2 : PetDog \sqsubseteq \exists 2 : hates.(1 : Cat)$

where $1 : Dog$ and $1 : Cat$ are defined in $\mathbf{P_{Animal}}$ but are imported into $\mathbf{P_{Pet}}$. The example also shows P-DL can represent both inter-module concept subsumptions (as DDL does) and inter-module role relations.

The package extension to DL is denoted as $\mathcal{P}$. For example, $\mathcal{ALCP}$ is the package-based version of DL $\mathcal{ALC}$. $\mathcal{P}_\mathcal{C}$ denotes a restricted type of package extension which only allows acyclic import of concept names. An XML syntax of P-DL to connect OWL ontologies, P-OWL, is under design.

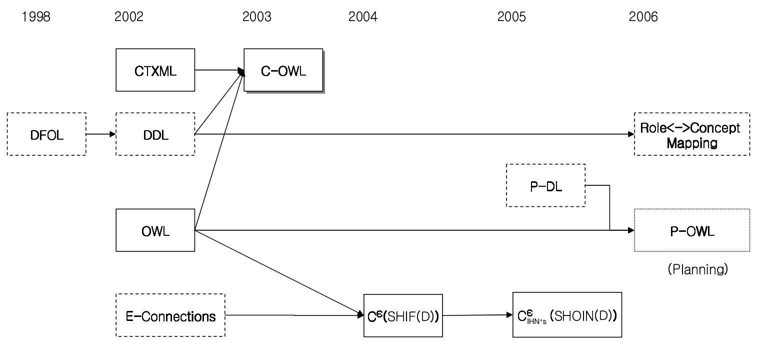

Fig. 1. Evolution of Modular Ontology Languages

The evolution of modular ontology language proposals is summarized in Figure 1.

3 Semantics of Modular Ontology Languages

3.1 Semantics of DDL, $\mathcal{E}$-Connections and P-DL

Although various modular ontology language proposals differ from each other in terms of language features, they share one feature in common in contrast with traditional ontology languages: all of the modular ontology language proposals support localized semantics. In other words, a (global) model for a modular ontology would contain a set of local models as well as a set of relations between those local models. In contrast, a traditional ontology (as well as fusion of logics [3]) always requires a single model that satisfies all restrictions in that ontology.

Serafini et.al. [34] and Bao et.al. [5] compared the semantics of different modular ontology language proposals in the light of DFOL semantics. This paper will follow their approach and further investigate some important properties of modular ontology semantics, such as local domain disjointness.

Formally, an abstract modular ontology (AMO) language is a DFOL knowledge base consisting of a family of component languages $\{L_i\}$ (each called a module) and semantic relations $\{M_{ij}\}(i \neq j)$. Each L_i is a subset of description logics (DL). In this paper, we restrict ourself to the setting where each L_i is a subset of the expressive DL $\mathcal{SHOIQ}(D)$, which is the logic foundation for OWL-DL. The modular ontology language proposals differ mainly with respect to how to define and interpret semantic relations $\{M_{ij}\}$.

A model of AMO includes a set of local models $\{\mathcal{I}_i\}$ and domain relations $\{r_{ij}\}$. For each L_i, there exists a local model $\mathcal{I}_i = \langle \Delta_i, (.)_i \rangle$, where Δ_i is the local interpretation domain, $(.)_i$ is the assignment function for concept, role and individual terms in L_i. A semantic relation M_{ij} from L_i to L_j is interpreted as a *domain relation* $r_{ij} \subseteq \Delta_i \times \Delta_j$, where $i \neq j$. A domain relation r_{ij} represents the capability of the module j to map the objects of Δ_i into Δ_j. Note that it is possible to have multiple domain relations between a pair of local models. Finally, $r_{ij}(d)$ denotes the set $\{d' \in \Delta_j | \langle d, d' \rangle \in r_{ij}\}$. For a subset $D \subseteq \Delta_i$, $r_{ij}(D)$ denotes $\cup_{d \in D} r_{ij}(d)$.

Semantics of DDL, $\mathcal{E}$-Connections and P-DL are summarized in Table 1 and explained below:

DDL bridge rules include homogenous rules, which are concept-to-concept (defined in [11]) or role-to-role (defined in [14]) mappings, and heterogenous rules (defined in [17]), which are concept-to-role or role-to-concept mappings. There are specific types of domain relations rc_{ij} (for role to concept mapping) and cr_{ij} (concept to role mapping) to interpret heterogenous rules.

$\mathcal{E}$-connections allow construction of new concepts using links. For a link E from module i to module j, its interpretation $r_E \subseteq \Delta_i \times \Delta_j$ is a domain relation for E. Thus, the distributed model of an $\mathcal{E}$-connected ontology may have multiple domain relations between two local models. Such a semantics for links is also equivalent to allow an i-role to have the range from and only from Δ_j. Thus, link constructors, like inversion or inclusion, are different from role constructors that bridge roles in *different* modules (which are illegal in $\mathcal{E}$-connections).

All concepts constructed using a link E from i to j are i-concepts. Thus they can be used in i as other local i-concepts. For example, the axiom $i : PerOwner \sqsubseteq \exists owns.(j : Pet)$ would indicate a restriction in Δ_i such that:

$PerOwner^{\mathcal{I}_i} \subseteq r^{-}_{owns}(Pet^{\mathcal{I}_j}) \subseteq \Delta_i$

Note that the semantics of $\mathcal{E}$-connections given in Table 1 is strictly equivalent to the forms given in [24, 21] and [34] although our notation is more general to represent semantic relations other than subsumptions. For example, a concept intersection $\exists owns.(j : Pet) \sqcap i : Man$ can be interpreted as

$r^{-}_{owns}(Pet^{\mathcal{I}_j}) \cap Man^{\mathcal{I}_i}$

Proposal	**Semantic Mapping**	**Syntax**	**Semantics**
DDL	Homogenous INTO	$i : \phi \xrightarrow{\sqsubseteq} j : \psi$	$r_{ij}(\phi^{\mathcal{I}_i}) \subseteq \psi^{\mathcal{I}_j}$
	Homogenous ONTO	$i : \phi \xrightarrow{\sqsupseteq} j : \psi$	$r_{ij}(\phi^{\mathcal{I}_i}) \supseteq \psi^{\mathcal{I}_j}$
	Heterogenous concept/role INTO	$i : C \xrightarrow{\sqsubseteq} j : R$	$cr_{ij}(C^{\mathcal{I}_i}) \subseteq R^{\mathcal{I}_j}$
	Heterogenous concept/role ONTO	$i : C \xrightarrow{\sqsupseteq} j : R$	$cr_{ij}(C^{\mathcal{I}_i}) \supseteq R^{\mathcal{I}_j}$
	Heterogenous role/concept INTO	$i : P \xrightarrow{\sqsubseteq} j : D$	$rc_{ij}(P^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_j}$
	Heterogenous role/concept ONTO	$i : P \xrightarrow{\sqsupseteq} j : D$	$rc_{ij}(P^{\mathcal{I}_i}) \supseteq D^{\mathcal{I}_j}$
	Partial individual correspondence	$i : x \mapsto j : y$	$y^{\mathcal{I}_j} \in r_{ij}(x^{\mathcal{I}_i})$
	Complete individual correspondence	$i : x \overset{=}{\mapsto} j : \{y_1, y_2, ...\}$	$r_{ij}(x^{\mathcal{I}_i}) = \{y_1^{\mathcal{I}_j}, y_2^{\mathcal{I}_j}, ...\}$
$\mathcal{E}$-Connections	Existential Link Restriction	$i : (\exists E.(j : D))$	$r_E^-(D^{\mathcal{I}_j})$
	Universal Link Restriction	$i : (\forall E.(j : D))$	$\Delta_i \backslash r_E^-(\Delta_j \backslash D^{\mathcal{I}_j})$
	Number Link Restriction	$i : (\geqslant nE.D)$	$\lvert r_E(x) \cap D^{\mathcal{I}_j}\rvert_{\neq} \geqslant n$, for $\forall x \in (\geqslant nE.D)^{\mathcal{I}_i}$
		$i : (\leqslant nE.D)$	$\lvert r_E(x) \cap D^{\mathcal{I}_j}\rvert_{\neq} \leqslant n$, for $\forall x \in (\leqslant nE.D)^{\mathcal{I}_i}$
	Link Inverse	$E = F^-$	$r_E = r_F^-$
	Link Inclusion	$E_1 \sqsubseteq E_2$	$r_{E_1} \subseteq r_{E_2}$
	Transitive Link	$Trans(G; I)$	$(x, y) \in r_{G(ij)} \wedge (y, z) \in r_{G(jk)} \rightarrow (x, z) \in r_G$, for $i, j, k \in I$
	Symmetric Link	$Symmetric(G)$	$r_{G(ij)} = r^-_{G(ji)}$
P-DL	importing	$i \xrightarrow{\phi} j$	$\phi^{\mathcal{I}_i} = \phi^{\mathcal{I}_j}$

Notations:

- DDL: ϕ is an i-concept(or role), ψ is an j-concept(or role); C is an i-concept, D is a j-concept, P is an i-role, R is a j-role; x is an i-individual, y_i is a j-individual; $r_{ij} \subseteq \Delta_i \times \Delta_j$ is the domain relation from i to j; $rc_{ij} \subseteq \Delta_i \times \Delta_i \times \Delta_j$ is the role to concept domain relation, $cr_{ij} \subseteq \Delta_i \times \Delta_j \times \Delta_j$ is the concept to role domain relation.
- E-Connections: E, E_1, E_2 is an $\mathcal{E}$-connection from i to j, F is an $\mathcal{E}$-connection from j to i; D is a j-concept; G is a generalized link; I is a set of module indices; r_E is the domain relation for E, r_E^- is the inverse of r_E; $\lvert S\rvert_{\neq}$ stands for all-different cardinality of a set S, i.e. the number of elements in S if equivalent elements only counted as one element.
- P-DL: ϕ is a concept, property or individual name.

Table 1. Semantics of Modular Ontology Languages

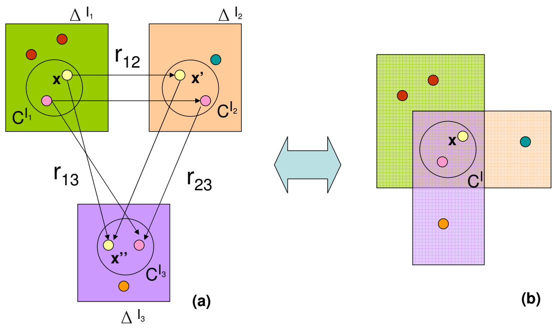

Fig. 2. P-DL Semantics (a) partially overlapping local models (b) virtual global model

$i : (\forall E.(j : D))$ is interpreted as $\Delta_i \backslash r_E^-(\Delta_j \backslash D^{\mathcal{I}_j})$ since $\forall E.D = \neg\exists E.(\neg D)$.

Transitive and symmetric links are generalized links [19, 30] to work around the strict local domain disjointness while still have expressive links in a certain sense. The basic idea is using "punning", i.e., allowing a link being interpreted in different contexts, and each of the interpretation is denoted with a superscript. For example, a link G may be used as $G^{(1)}$ from i to j and $G^{(2)}$ from j to k; G's interpretation will be the union of $r_{G^{(1)}} \subseteq \Delta_i \times \Delta_j$ and $r_{G^{(2)}} \subseteq \Delta_j \times \Delta_k$.

P-DL uses importing relations to connect local models. In contrast to OWL, which forces the model of an imported ontology be completely embedded in a global model, the P-DL importing relation is *partial* in that only commonly shared terms are interpreted in the overlapping part of local models. It can also be expressed using AMO domain relations: the *image domain relation* between $\mathcal{I}_i$ and $\mathcal{I}_j$ is $r_{ij} \subseteq \Delta_i \times \Delta_j$. P-DL importing relation is:

– one-to-one: for any $x \in \Delta_i$, there is at most one $y \in \Delta_j$, such that $(x, y) \in r_{ij}$, and vice versa.
– compositionally consistent: $r_{ij} = r_{ik} \circ r_{jk}$, where $\circ$ denotes function composition. Therefore, semantic relations between terms in i and terms in k can be inferred even if k doesn't directly import terms from i.

Thus, a P-DL model is a virtual model constructed from partially overlapping local models by merging "shared" individuals, as shown in Figure 2.

Using the AMO framework representation of the three major modular ontology language proposals, in what follows we will discuss an important problems in their semantics: whether local domains should be disjoint.

3.2 Local Domain Disjointness

$\mathcal{E}$-connections explicitly requires that local domains of each module should be strictly disjoint. The main reason for such a restriction is to ensure decidability of the whole ontology if each module is already locally decidable. However, as mentioned earlier, such a restriction also seriously limits the expressivity of $\mathcal{E}$-Connections as well as the capacity to preform some reasoning tasks [5]. For example, it is not possible in $\mathcal{E}$-Connections to infer if a concept $i : C$ is more general than a concept $j : D$ $(i \neq j)$ in another module, while such a inference task is one of the most common types of inference needed in many practical applications.

Local domain disjointness is not explicitly required in DDL semantics. On the other hand, DDL semantics is also neutral to such a possibility, i.e., it does not try to utilize such information. For example, if an individual x is shared by two DDL local domains Δ_1 and Δ_2, i.e., $x \in \Delta_1 \cap \Delta_2$, $1 : x$ and $2 : x$ are not treated as the same individual in a DDL model. The relation between them has to be established also by the domain relation r_{12}, such as $(1 : x, 2 : x) \in r_{12}$. However, since DDL domain relations do not indicate individual identity (actually such an identity semantics is avoided on purpose in DDL), it is also possible to map $1 : x$ to other individuals from local domain 2, e.g. $2 : y$. Thus, even if two individuals $1 : x$ and $2 : x$ are identifiers for the same object in the physical world, they are still treated *as if* they are different individuals in DDL. Therefore, we believe DDL in effect, forces local domains to be disjoint implicitly.

DDL's avoidance of modelling domain relations in terms of individual identity is intended to allow loose coupling of local domains and more expressive individual correspondences between local domains [11], for example, $i : NYC \overset{=}{\mapsto} j : \{Bronx, Manhattan, Queens, Brooklyn, StatenIsland\}$. However, such a mechanism may also lead to inference difficulties.

While concept bridge rules are intended to simulate concept subsumptions, their semantic behaviors are still different in many scenarios. For instance, $i : C \xrightarrow{\sqsubseteq} j : D$ may not have the same semantics as typically desired, i.e. C is less general than D, since $r_{ij}(C^{\mathcal{I}_i})$ may be $\emptyset$ (empty set) thus is a subset of any $D^{\mathcal{I}_j}$; it is different from a concept subsumption $C \sqsubseteq D$ where a non-empty set $C^{\mathcal{I}}$ (when C is satisfiable) must be a subset of of $D^{\mathcal{I}}$. If the domain relation is not injective, unsatisfiability may not be preserved across DDL modules. For example, $1 : Bird \xrightarrow{\sqsupseteq} 2 : Penguin$ and $\neg 1 : Fly \xrightarrow{\sqsupseteq} 2 : Penguin$ do not render $2 : Penguin$ unsatisfiable even if L_1 entails $Bird \sqsubseteq Fly$.

Since a domain relation r_{ij} represents the subjective point of view of the module j, in general, it is not transitively reusable (in contrast to individual identity relations that are transitive). For example, $(x, y) \in r_{12}$ and $(y, z) \in r_{23}$ does not automatically indicate $(x, z) \in r_{13}$, thus $1 : Bird \xrightarrow{\sqsupseteq} 2 : Fowl$ and $2 : Fowl \xrightarrow{\sqsupseteq} 3 : Chicken$, do not in general ensure that $1 : Bird \xrightarrow{\sqsupseteq} 3 : Chicken$. This might present a difficulty if we want to reuse knowledge from *indirectly* connected modules.

Thus, since the local domain disjointness presents some serious expressivity limitations as well as inference difficulties, it is natural to ask: "Can we relax such an assumption while still have desirable semantic features (e.g. decidability)"? Complete overlap between local models, as required by OWL, are not desirable since it requires an integrated ontology for reasoning. Thus, a practical approach would be to allow local domains that are only *partially overlapping*.

One of the first efforts in this direction is the inter-schema terminology mapping mechanism proposed by Catarci and Lenzerini [15]. Their framework defines mappings between concepts in forms of

$i : C \sqsubseteq_{ext} j : D$, semantics $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

$i : C \sqsubseteq_{int} j : D$, semantics $C^{\mathcal{I}} \cap \Delta_i \cap \Delta_j \subseteq D^{\mathcal{I}} \cap \Delta_i \cap \Delta_j$

where the global domain is the union of multiple local domains $\Delta_1, \Delta_2, ...$; two local domains may be partially overlapping, i.e. $\Delta_i \cap \Delta_j \neq \emptyset$, for some $i \neq j$.

P-DL provides a more expressive mechanism to connect partially overlapping local models. It is easy to reduce DDL bridge rules between concepts and $\mathcal{E}$-connection links to P-DL $\mathcal{P}_{\mathcal{C}}$ expressions. A bridge rule $i : C \xrightarrow{\sqsubseteq} j : D$ or $i : C \xrightarrow{\sqsupseteq} j : D$ can be reduced to a subsumption in j with C as an imported term; an $\mathcal{E}$-Connections link E (from i to j) can be defined as an i-role with $\top_j$ in its range. Link constructors can be reduced to local role constructors in this fashion. If we use the more expressive $\mathcal{P}$ extension (i.e. allowing role importing), we can also express DDL bridge rules between roles and transitive/symmertic links in $\mathcal{E}$-Connections.

The DDL features that can not be directly reduced to P-DL expressions are DDL heterogenous bridge rules and individual correspondences. In general, DDL heterogenous bridge rules are not directly reducible to DL $\mathcal{SHOIQ}(D)$ axioms since they involve ternary first order predicates (while $\mathcal{SHOIQ}(D)$ can be reduced to FOL with binary predicates). Individual correspondences in P-DL require the importing of nominal concepts (individual names) across ontology modules. However, at present, OWL-DL (i.e. $\mathcal{SHOIQ}(D)$), does not allow subsumption relations between nominal concepts. Thus the package-extended version of OWL-DL, i.e., $\mathcal{SHOIQP}(D)$, can only express one-to-one individual correspondence in the form of $i : x = j : y$. Nevertheless, since DDL domain relations do not indicate identity relations, we believe it is better to translate many-to-one and one-to-many individual correspondences using roles (since domain relations can be modelled as roles [11]). For example, to translate complete individual correspondence

$i : NYC \overset{=}{\mapsto} j : \{Bronx, Manhattan, Queens, Brooklyn, StatenIsland\}$

We may define a new role $j : contains$, a new nominal concept $NycBoroughs$ with enumerated members $Bronx, Manhattan, Queens, Brooklyn, StatenIsland$, and a new axiom in j:

$i : NYC \equiv \forall j : contains.(j : NycBoroughs)$

Thus, by relaxing the local domain disjointness assumption, we can obtain the expressivity offered by both DDL and $\mathcal{E}$-Connections in P-DL. Furthermore, P-DL extension $\mathcal{P}$ may also support inter-module role constructors that are not supported by either DDL or $\mathcal{E}$-connections, such as intersection $i : P \sqcap j : Q$, with semantics $P^{\mathcal{I}} \subseteq \Delta_i \times \Delta_i$, $Q^{\mathcal{I}} \subseteq \Delta_j \times \Delta_j$, and $P^{\mathcal{I}} \cap Q^{\mathcal{I}} \subseteq (\Delta_i \cap \Delta_j) \times (\Delta_i \cap \Delta_j)$ (note that this is not the same as $\mathcal{E}$-connections link intersetion where two links P, Q should be both from the domain of $\Delta_i \times \Delta_j$ and $\Delta_i \cap \Delta_j = \emptyset$).

Another point of concern is that if partially overlapping local domain semantics can ensure decidability if the individual ontology modules were decidable. In general, the union of two decidable fragments of DL may be undecidable [3, 25]. Fortunately, in a setting as 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.

Furthermore, since overlapped partial domains provided an unambiguous communication avenue among multiple modules, it can be ensured that conclusions drawn form reasoning with a P-DL ontology is always same as that obtained from the reasoning with an integrated ontology [5, 6]. Thus many inference difficulties presented in DDL and $\mathcal{E}$-connections can be avoided. A sound and complete distributed reasoning algorithm for the P-DL $\mathcal{ALCP}_{\mathcal{C}}$ proposed in [7] supports reasoning over multiple ontology modules using messages between modules (without the need to combine the modules into a single ontology). In short, we believe that relaxing the local domain disjointness assumption can significantly improve the expressivity of the modular ontology language without harming the decidability of the language in the setting of web ontologies.

4 Conclusions

Contribution of the paper includes the following: we reviewed the evolution of modular ontology languages and compared several recent language proposals for modular ontologies on their language features and semantics; we presented a new AMO-based representation for $\mathcal{E}$-Connections that is capable of expressing semantic relations across ontology modules other than subsumptions studied in [25, 21, 34, 5]; we showed that relaxing the local domain disjointness assumption can improve the expressivity of modular languages while avoiding some of the semantic difficulties present in current proposals; we also showed how to realize some DDL and $\mathcal{E}$-Connection features into P-DL; by allowing role and individual importing in P-DL, we have extended the P-DL expressivity beyond that presented in [5].

While there has been extensive efforts on modular ontology languages during the past 4 years, many problems still need to be addressed before modular ontology languages can be used in large-scale semantic web applications:

- In contrast to classical DL, where there is well-understood studies on the decidability, complexity of expressive DL languages, there will lack a comprehensive understanding in modular ontology languages on those issues. For example, what is the maximal set of inter-module formulas that ensures decidability of the obtained language [3]? What is the complexity upper bound of reasoning procedures of expressive modular ontology languages[4]?
- A consensus on an OWL-compatible syntax for a modular ontology language that can express both inter-module concept subsumptions and inter-module role relations is still lacking. It would be interesting to investigate whether OWL can be re-modelled with a new modular semantics, or it has to be extended with a new set of constructors to replace `owl:imports`
- Existing ontology reasoners need to be extended to support modular ontologies. It would be interesting to explore extending reasoners such as DRAGO [36] or Pellet [37], to support more expressive modular ontology languages, as well as avoid a materialized global model in a single memory place (otherwise the reasoning process can be done by a classic reasoner on an integrated ontology)

[3] F. Baader and S. Ghilard studied conditions under which decidability of the validity of universal formulae in component many-sorted theories transfers to their connection [2]. It generalizes the decidability of DDL and $\mathcal{E}$-Connections since formulae in [2] are not necessarily only unary (thus it is also applicable to DL roles (binary FOL predicates)). However, there still lacks similar result for existential qualified formulae.

[4] Some special cases are studied. Connections of many-sorted theories [2] renders NEXPTIME and $\mathcal{E}$-Connections [19] gives 2NEXPTIME results that are both higher than the complexity of the decision procedures for the component logics (EXPTIME). Bao et. al. show that in [4, 7] a restricted version of P-DL, $\mathcal{ALCP}_C$, has a decision procedure that has no higher complexity than that of component logics (EXPTIME-complete).

– Mature tools for building modular ontologies are needed. COB Editor [9] [5] offers editing modular ontologies collaboratively where each module has DAG hierarchy as skeleton (within the expressivity of OBO-format ontologies). Tools (such as a Protege plugin) that support more expressive modular ontology languages still need to be explored.

Acknowledgement: This research was supported in part by a grant from the US NSF (IIS-0639230) and support from the USDA Bioinformatics Coordination Project.

References

1. E. Amir and S. A. McIlraith. Partition-based logical reasoning. In *KR*, pages 389–400, 2000.
2. F. Baader and S. Ghilardi. Connecting many-sorted theories. In *CADE*, pages 278–294, 2005.
3. F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics. In *Description Logics*, pages 21–30, 2000.
4. J. Bao, D. Caragea, and V. Honavar. A distributed tableau algorithm for package-based description logics. In *the 2nd International Workshop On Context Representation And Reasoning (CRR 2006), co-located with ECAI 2006*. 2006.
5. 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.
6. J. Bao, D. Caragea, and V. Honavar. On the semantics of linking and importing in modular ontologies. In *I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273*, pages 72–86. 2006.
7. J. Bao, D. Caragea, and V. Honavar. A tableau-based federated reasoning algorithm for modular ontologies. In *Accepted by 2006 IEEE/WIC/ACM International Conference on Web Intelligence (In Press)*, 2006.
8. 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.
9. J. Bao, Z. Hu, D. Caragea, J. Reecy, and V. Honavar. Developing frameworks and tools for collaborative building of large biological ontologies. In *the 4th International Workshop on Biological Data Management (BIDM'06), @ DEXA'06*. 2006.
10. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
11. A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In *CoopIS*, pages 36–53, 2002.
12. A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184, 2003.
13. P. Bouquet, A. Dona, L. Serafini, and S. Zanobini. Conceptualized local ontologies specification via CTXML. In *AAAI-02 Workshop on Meaning Negotiation (MeaN-02) July 28, 2002, Edmonton, Canada*, 2002.
14. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-owl: Contextualizing ontologies. In D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003.
15. T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. In *CoopIS*, pages 55–62, 1993.

[5] http://sourceforge.net/projects/cob/

16. 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.
17. C. Ghidini and L. Serafini. Reconciling concepts and relations in heterogeneous ontologies. In *ESWC*, pages 50–64, 2006.
18. F. Giunchiglia and C. Ghidini. Local models semantics, or contextual reasoning = locality + compatibility. In *KR*, pages 282–291, 1998.
19. B. C. Grau. *Combination and Integration of Ontologies on the Semantic Web*. PhD thesis, Dpto. de Informatica, Universitat de Valencia, Spain, 2005.
20. 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.
21. 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.
22. B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *KR2006*, 2006.
23. I. Horrocks. The FaCT system. In *Tableaux*, pages 307–312, 1998.
24. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of description logics. In *Description Logics Workshop, CEUR-WS Vol 81*, 2003.
25. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.
26. O. Kutz, F. Wolter, and M. Zakharyaschev. Connecting abstract description systems. In *KR*, pages 215–226, 2002.
27. D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):32–38, 1995.
28. Y. Ma and J. Wei. A default extension to distributed description logics. In *IAT*, pages 38–44, 2004.
29. Y. Ma, J. Wei, B. Jin, and S. Liu. A formal framework for ontology integration based on a default extension to ddl. In *ICTAC*, pages 154–169, 2004.
30. B. Parsia and B. C. Grau. Generalized link properties for expressive epsilon-connections of description logics. In *AAAI*, pages 657–662, 2005.
31. 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.
32. A. L. Rector, C. Wroe, J. Rogers, and A. Roberts. Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies. In *K-CAP*, pages 139–146, 2001.
33. L. Serafini, A. Borgida, and A. Tamilin. Aspects of distributed and modular ontology reasoning. In *IJCAI*, pages 570–575, 2005.
34. L. Serafini, H. Stickenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In *19th IJCAI*, 2005.
35. L. Serafini and A. Tamilin. Distributed instance retrieval in heterogeneous ontologies. In *Proceedings of SWAP 2005, CEUR Workshop Vol 166*, 2005.
36. L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376, 2005.
37. E. Sirin and B. Parsia. Pellet: An OWL DL Reasoner. In *Description Logics Workshop*, 2004.
38. H. Stuckenschmidt and M. Klein. Modularization of ontologies - wonderweb: Ontology infrastructure for the semantic web, ist project 2001-33052, version 1.0.
39. H. Stuckenschmidt and M. C. A. Klein. Structure-based partitioning of large concept hierarchies. In *International Semantic Web Conference*, pages 289–303, 2004.