Modular Ontology Languages Revisited

Bernardo Cuenca-Grau and Oliver Kutz
The University of Manchester
School of Computer Science
Manchester, M13 9PL, UK
{bcg,ok}@cs.man.ac.uk

Abstract

In this paper, we compare various formalisms that have been recently introduced or used for distributed reasoning, ontology integration, and related topics; in particular, we focus on $\mathcal{E}$-connections, Distributed Description Logics, and Package-based Description Logics. We then establish the relationship between these formalisms and various non-standard reasoning services that have been recently proposed for assisting the modeler in ontology integration and knowledge reuse tasks, such as locality of an ontology and conservative extensions.

1 Introduction

The design and integration of ontologies formulated in modern ontology languages, such as OWL, is a serious challenge. In particular, *modularity* is a key requirement for many aspects of ontology design, maintenance and integration. Modular representations are easier to understand, reason with, debug, extend and reuse.

In contrast to other disciplines such as software engineering, in which modularity is a well established notion, the problem of formally characterising a modular representation for ontologies is not yet well-understood.

Achieving a reasonable notion of modularity for ontologies is crucial for assisting the ontology designer in numerous key tasks:

- *Collaborative ontology development.* Very large ontologies, such as SNOMED and the NCI Thesaurus, are not created and maintained by a single person. The development of these ontologies generally involves a team of experts, which often need to communicate and reconcile their changes. In order to minimise the interaction between modelers and facilitate maintenance, the changes performed by a modeler should not have an impact in other parts of the ontology under the control of other modelers.
- *Partial knowledge reuse.* For large ontologies, it is crucial that there are extractable parts that can be reused outside the context of the original ontology. Those fragments should not be arbitrary in the sense that they should preserve key aspects of their meaning in the original ontology.
- *Controlled Ontology Integration.* In applications involving multiple ontologies, these ontologies are often not completely independent. The most straightforward way to integrate a set of ontologies is to simply take the union of their axioms. A more sophisticated integration typically involves the establishment of mappings that glue the ontologies together by relating the meaning of different symbols in the different ontologies. In any case, the relationship between the semantics of the combination and their parts should be controlled and well-understood.
- *Efficient modular reasoning.* Although modern reasoners perform well in realistic ontologies, reasoning with large ontologies is often still hard in practice. Even if the ontology under consideration can be processed, it may still be the case that the processing time involved is too high for ontology engineering applications, which may require a fast response under changes in the ontology. A good decomposition of a large ontology into modules may be crucial to select the part of the ontology that is sufficient to answer a query, or that is affected by a change in the ontology.

In the last few years, an increasing body of work has developed in the direction of establishing a plausible notion of modularity for ontologies in order to assist the ontology engineer in performing these tasks. We distinguish two approaches in the literature:

1. the design of *formalisms* that provide control over the interaction between the modules, [Borgida and Serafini, 2003; Kutz *et al.*, 2004; Bao *et al.*, 2006c; Cuenca-Grau *et al.*, 2006b], and
2. the design of specialised non-standard *reasoning services* [Cuenca-Grau *et al.*, 2006c; Ghilardi *et al.*, 2006; Cuenca-Grau *et al.*, 2006a; Lutz *et al.*, 2007; Cuenca-Grau *et al.*, 2007].

In the former approach, a module is represented as a component of a global setting (with a ‘local’ language and a ‘local’ semantics). The formalism then provides new syntax, with its corresponding semantics, to model the interaction between the modules. In this paper, we refer to these languages as

Modular Ontology Languages (or MOL for short) and focus, in particular, on the relationship between three formalisms:

- $\mathcal{E}$-Connections [Kutz *et al.*, 2004; Cuenca-Grau *et al.*, 2006b]
- Distributed Description Logics (DDL) [Borgida and Serafini, 2003]
- Package-based Description Logics (P-DL) [Bao *et al.*, 2006c; 2006b]

A rather different approach for supporting the tasks mentioned above is to establish a set of non-standard reasoning services defined over the ontologies $\mathcal{T}_1, \ldots, \mathcal{T}_n$ under consideration or over the union $\mathcal{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_n$ of their axioms. The aim of these services is to test different aspects of the relationship between the semantics of the union ontology $\mathcal{T}$ and the semantics of their components $\mathcal{T}_i$. This approach does not assume the existence of a combination language with a specialised semantics, but may require the development of new reasoning algorithms for the new services.

Our contributions in this paper are as follows. Firstly, we investigate the relationship between $\mathcal{E}$-connections, DDL and P-DL, by comparing their expressive power and computational properties. Secondly, we consider two recently proposed non-standard reasoning services, namely *locality* [Cuenca-Grau *et al.*, 2006c] of a TBox, and *conservative extensions* [Lutz *et al.*, 2007] in the context of ontologies, and we establish the relationship between $\mathcal{E}$-connections and DDL with these reasoning services.

The proofs of all new results presented in this paper can be found in an accompanying technical report [Cuenca-Grau and Kutz, 2006].

We begin by defining $\mathcal{E}$-connections, show that they do not presuppose disjoint domains, give a reduction of $\mathcal{E}$-connection reasoning to reasoning in a single DL, and briefly study their expressivity.

2 $\mathcal{E}$-connections of DLs

$\mathcal{E}$-connections were originally conceived as a versatile and well-behaved technique for combining logics [Kutz *et al.*, 2004; Kutz, 2004], but have been quickly adopted as a framework for the integration of ontologies and modular reasoning in the Semantic Web [Cuenca-Grau *et al.*, 2006b]. The general idea behind this combination method is that the interpretation domains of the connected logics are interpreted by disjoint vocabulary and interconnected by means of *link relations*. The language of the $\mathcal{E}$-connection is then the union of the original languages enriched with operators capable of talking about the link relations.

The most important feature of $\mathcal{E}$-connections is that, just as DLs themselves, they offer an appealing compromise between expressive power and computational complexity: although powerful enough to express many interesting concepts, the coupling between the combined logics is sufficiently loose for proving general results about the transfer of decidability. Such transfer results state that if the connected logics are decidable, then their connection will also be decidable. In this paper, we define $\mathcal{E}$-connections of DLs only.[1]

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two DLs that are to be connected.[2] We assume that the **languages** $\mathcal{L}_1$ and $\mathcal{L}_2$, i.e., the concept, role, and object names of $\mathcal{S}_1$ and $\mathcal{S}_2$, are pairwise disjoint. To form a connection, fix a non-empty set $\mathcal{E} = \{E_j \mid j \in J\}$ of binary relation symbols. The set of **concepts** of the **basic $\mathcal{E}$-connection language** for $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ is partitioned into a set of 1-concepts and a set of 2-concepts. Intuitively, i-concepts are the concepts of $\mathcal{L}_i$ enriched with new concept constructors for talking about link relations. We often refer to an $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ simply as $\mathcal{C}^{\mathcal{E}}$ once the $\mathcal{S}_i$ have been fixed. In the following, we set $\overline{1} = 2$ and $\overline{2} = 1$ and denote by $|A|$ the cardinality of a set A.

Definition 1 *The sets of* 1-concepts *and* 2-concepts *of* $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ *are defined by simultaneous induction: for* $i \in \{1, 2\}$,

(1) *every concept name of* $\mathcal{L}_i$ *is an* i*-concept;*

(2) *the set of* i*-concepts is closed under* $\neg$, $\wedge$, *and the concept constructors of* $\mathcal{L}_i$*;*

(3) *if* C *is an* $\overline{i}$*-concept, then the expression* $\langle E_j \rangle^i C$ *is an* i*-concept, for every* $j \in J$.

The set of i***-concepts*** *of* $\mathcal{C}^{\mathcal{E}}$ *is denoted by* $\mathsf{Con}_i(\mathcal{C}^{\mathcal{E}})$, $i = 1, 2$. *The set of* ***concepts*** *of* $\mathcal{C}^{\mathcal{E}}$ *is* $\mathsf{Con}_1(\mathcal{C}^{\mathcal{E}}) \cup \mathsf{Con}_2(\mathcal{C}^{\mathcal{E}})$. *The* ***concept assertions*** *of* $\mathcal{C}^{\mathcal{E}}$ *are of the form* $C_1 \sqsubseteq C_2$, *where* $C_1, C_2 \in \mathsf{Con}_i(\mathcal{C}^{\mathcal{E}})$, $i \in \{1, 2\}$. *The* ***object assertions*** *of* $\mathcal{C}^{\mathcal{E}}$ *are of the form* aE_jb, *where* a *is a* 1*-object,* b *is a* 2*-object, and* $E_j \in \mathcal{E}$. *A* ***knowledge base*** *is a triple* $\Gamma = \langle \Gamma_1, \Gamma_2, \Gamma_o \rangle$, *where* Γ_i, $i = 1, 2$, *are finite sets of* i*-concept assertions, and* Γ_o *is a finite set of object assertions.*

If $|\mathcal{E}| \in \mathbb{N}$, we say that $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ is **finitely linked**. If $|\mathcal{E}| = 1$, we say that $\mathcal{C}$ is **unarily linked**.

A strictly weaker language, closer in spirit to DDLs, is defined as follows. The language of **one-way** $\mathcal{E}$-connections for an $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ is defined by assuming that

$$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2, \text{ with } \mathcal{E}_1 = \{E_i^{12} \mid i \in I_1\}, \mathcal{E}_2 = \{E_j^{21} \mid j \in I_2\}$$

and by replacing item (3) of Definition 1 by

(3.1) if C is a 2-concept and $i \in I_1$, then the expression $\langle E_i^{12} \rangle C$ is a 1-concept.

(3.2) if D is a 1-concept and $j \in I_2$, then the expression $\langle E_j^{21} \rangle D$ is a 2-concept.

[1] The generality of the transfer results for $\mathcal{E}$-connections obtained in [Kutz *et al.*, 2004; Kutz, 2004] is due to the fact that $\mathcal{E}$-connections are defined and investigated using the framework of so-called *abstract description systems* (ADSs), a common generalisation of description logics, modal logics, logics of time and space, and many other logical formalisms [Baader *et al.*, 2002]. Thus, we can connect not only DLs with DLs, but also, say, description logics with spatial logics. A natural interpretation of link relations in this context would then be, for instance, to describe the spatial extension of abstract (DL) objects. Moreover, several extensions to the basic $\mathcal{E}$-connection language have been studied in [Kutz *et al.*, 2004], including Booleans on links, number restrictions on links, link operators on object names, and first-order link constraints.

[2] In general, $\mathcal{E}$-connections can connect n ADSs for any $n \in \mathbb{N}$, and all the formulated results apply to the n-dimensional case as well [Kutz *et al.*, 2004].

A one-way $\mathcal{E}$-connection is **finitely linked** if $|I_1|, |I_2| \in \mathbb{N}$, and **unarily linked** if $|I_1| = |I_2| = 1$.

As expected, a model for the $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ consists of a model for $\mathcal{S}_1$, a model for $\mathcal{S}_2$, and an interpretation of the link relations.

Definition 2 [$\mathcal{E}$-connection Semantics] *A structure*

$$\mathfrak{M} = \left\langle \mathfrak{W}_1, \mathfrak{W}_2, \mathcal{E}^{\mathfrak{M}} = (E_j^{\mathfrak{M}})_{j \in J} \right\rangle,$$

where $\mathfrak{W}_i = (W_i, .^{\mathfrak{W}_i})$ is an interpretation of $\mathcal{S}_i$ for $i \in \{1,2\}$ and $E_j^{\mathfrak{M}} \subseteq W_1 \times W_2$ for each $j \in J$, is called an ***interpretation*** *for $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$. The* **extension** *$C^{\mathfrak{M}} \subseteq W_i$ of an i-concept C is defined by simultaneous induction. For concept names C of $\mathcal{L}_i$, we put $C^{\mathfrak{M}} = C^{\mathfrak{W}_i}$; the inductive steps for the Booleans and function symbols of $\mathcal{L}_i$ are standard; finally,*

$$(\langle E_j \rangle^1 C)^{\mathfrak{M}} = \{x \in W_1 \mid \exists y \in C^{\mathfrak{M}}\ (x,y) \in E_j^{\mathfrak{M}}\},$$
$$(\langle E_j \rangle^2 D)^{\mathfrak{M}} = \{x \in W_2 \mid \exists y \in D^{\mathfrak{M}}\ (y,x) \in E_j^{\mathfrak{M}}\}.$$

For object assertions we have

$$\mathfrak{M} \models aE_j b \iff (a^{\mathfrak{W}_1}, b^{\mathfrak{W}_2}) \in E_j^{\mathfrak{M}}.$$

The notion of ***truth*** *in an interpretation $\mathfrak{M}$,* ***satisfiability****, and* ***entailment****, can now be reduced to the standard notions for the component DLs in the obvious way.*

It is sometimes claimed that $\mathcal{E}$-connections make a strong assumption in requiring disjoint domains, and that this distinguishes them fundamentally from, e.g., DDLs or P-DLs [Bao *et al.*, 2006c; 2006b]. However, the formal definitions never made this assumption, rather, the assumption of disjoint domains was used to give a better intuition of the possible interactions between the different component DLs. A requirement of disjoint domains is not essential for the semantics. What is essential, however, is the disjointness of the *concept languages* of the component logics.

To make this obvious, let us define two variants of semantics of $\mathcal{E}$-connections. Call the semantics introduced in Definition 2 **free-floating** $\mathcal{E}$-connection semantics, and define **separated** $\mathcal{E}$-connection semantics by requiring additionally $W_1 \cap W_2 = \emptyset$.

Proposition 3 *An assertion is satisfiable w.r.t. separated $\mathcal{E}$-connection semantics if and only if it is satisfiable w.r.t. free-floating $\mathcal{E}$-connection semantics.*

We next show how reasoning in one-way $\mathcal{E}$-connections can be reduced to standard reasoning in a single DL, and also show how $\mathcal{E}$-connection semantics can be axiomatised within a DL Tbox.

2.1 Reduction of one-way $\mathcal{E}$-Connections to DL

Reasoning with one-way $\mathcal{E}$-connections of sufficiently expressive DLs can be reduced without loss of expressivity to reasoning in one component DL.[3]

Let $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ be an $\mathcal{E}$-connection based on disjoint $\mathcal{ALC}$ signatures $\mathsf{Sig}(\mathcal{S}_1)$ and $\mathsf{Sig}(\mathcal{S}_2)$. Let $\top_1$ and $\top_2$ be two *new* concept names, not appearing in $\mathsf{Sig}(\mathcal{S}_1) \cup \mathsf{Sig}(\mathcal{S}_2)$. Further, let $\mathcal{S}_3$ be the $\mathcal{ALC}$ DL built from symbols in $\mathsf{Sig}(\mathcal{S}_1) \cup \mathsf{Sig}(\mathcal{S}_2) \cup \mathcal{E} \cup \{\top_1, \top_2\}$, where the elements of $\mathcal{E}$ are treated as role names of $\mathcal{S}_3$. Denote by $\mathsf{Con}(\mathcal{S}_3)$ the set of (complex) concepts that can be constructed from $\mathsf{Sig}(\mathcal{S}_3)$,[4] and define a translation $.^\sharp : \mathsf{Con}(\mathcal{C}^{\mathcal{E}}) \longrightarrow \mathsf{Con}(\mathcal{S}_3)$, $i = 1,2$, as follows:

$$\begin{array}{rcll}
(A_i)^\sharp & := & A_i & , \quad A_i \text{ concept name of } \mathsf{Sig}(\mathcal{S}_i) \\
(C \sqcap D)^\sharp & := & C^\sharp \sqcap D^\sharp & , \quad C, D \in \mathsf{Con}_i(\mathcal{C}^{\mathcal{E}}) \\
(\neg C)^\sharp & := & \top_i \sqcap \neg C^\sharp & , \quad C \in \mathsf{Con}_i(\mathcal{C}^{\mathcal{E}}) \\
(\exists R.C)^\sharp & := & \exists R.C^\sharp & , \quad \exists R.C \in \mathsf{Con}_i(\mathcal{C}^{\mathcal{E}}) \\
(\langle E_i^{12} \rangle C)^\sharp & := & \exists E_i^{12}.C^\sharp & , \quad E_i^{12} \in \mathcal{E}_1, C \in \mathsf{Con}_2(\mathcal{C}^{\mathcal{E}}) \\
(\langle E_j^{21} \rangle D)^\sharp & := & \exists E_j^{21}.D^\sharp & , \quad E_j^{21} \in \mathcal{E}_2, D \in \mathsf{Con}_1(\mathcal{C}^{\mathcal{E}})
\end{array}$$

Note that this translation does not require $\top_1 \sqcap \top_2 \sqsubseteq \bot$. Moreover, the same translation with minor modifications can be carried out for basic $\mathcal{E}$-connections, but the target DL needs additionally inverse roles.

Given an $\mathcal{E}$-connection knowledge base $\Gamma = \langle \Gamma_1, \Gamma_2, \Gamma_o \rangle$, we can encode the structure of $\mathcal{E}$-connection models in an $\mathcal{ALC}$ Tbox as follows. For $i = 1,2$, $R^i, A^i \in \mathsf{Sig}(\Gamma_i)$, $E_j^{12} \in \mathcal{E}_1$, and $E_k^{21} \in \mathcal{E}_2$, define $\Gamma_{\mathcal{C}^{\mathcal{E}}}$ as the union of

$$\begin{array}{lcl}
\mathsf{dom}(E_i^{12}) \sqsubseteq \top_1 & , & \mathsf{range}(E_i^{12}) \sqsubseteq \top_2 \\
\mathsf{dom}(E_j^{21}) \sqsubseteq \top_2 & , & \mathsf{range}(E_j^{21}) \sqsubseteq \top_1 \\
\mathsf{dom}(R^i) \sqsubseteq \top_i & , & \mathsf{range}(R^i) \sqsubseteq \top_i \\
A_i \sqsubseteq \top_i & &
\end{array}$$

where, e.g., $\mathsf{dom}(R^i) \sqsubseteq \top_i$ is short for $\exists R^i.\top \sqsubseteq \top_i$ and $\mathsf{range}(R^i) \sqsubseteq \top_i$ is short for $\top \sqsubseteq \forall R^i.\top_i$.

Define the class $\mathfrak{M}_\Gamma$ of **bridged** $\mathcal{S}_3$-models for Γ by selecting all models $\mathcal{I}$ of $\mathcal{S}_3$ such that $\mathcal{I} \models \Gamma_{\mathcal{C}^{\mathcal{E}}}$. It should be clear that the class of models of $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ for $\mathsf{Sig}(\Gamma)$ and the class $\mathfrak{M}_\Gamma$ of bridged $\mathcal{ALC}$-models for Γ are isomorphic.

Given a knowledge base Γ for $\mathcal{C}^{\mathcal{E}}$, extend the translation $.^\sharp$ to object and concept assertions as follows

$$\begin{array}{rcl}
(aE_jb)^\sharp & := & aE_jb \\
(C \sqsubseteq D)^\sharp & := & C^\sharp \sqsubseteq D^\sharp
\end{array}$$

We now easily obtain:

Proposition 4 *For any $\mathcal{E}$-connection knowledge base Γ, any bridged $\mathcal{S}_3$-model $\mathcal{I} \in \mathfrak{M}_\Gamma$, and every $\mathcal{C}^{\mathcal{E}}$-assertion ϕ:*

$$\mathcal{I} \models \phi \iff \mathcal{I} \models \phi^\sharp.$$

Moreover, to translate an $\mathcal{E}$-connection knowledge base Γ together with its specialised semantics, we can set

$$\Gamma^{\mathcal{ALC}} := \Gamma_1^\sharp \cup \Gamma_2^\sharp \cup \Gamma_o^\sharp \cup \Gamma_{\mathcal{C}^{\mathcal{E}}}$$

to obtain

Proposition 5 *Γ is satisfiable w.r.t. free-floating $\mathcal{E}$-connection semantics if and only if $\Gamma^{\mathcal{ALC}}$ is satisfiable w.r.t. $\mathcal{ALC}$ semantics.*

[3] A similar reduction has been carried out for DDL reasoning in [Borgida and Serafini, 2003].

[4] Note that there are no restrictions assumed on the usage of symbols in $\mathsf{Sig}(\mathcal{S}_1) \cup \mathsf{Sig}(\mathcal{S}_2) \cup \mathcal{E} \cup \{\top_1, \top_2\}$ to build complex concept of $\mathcal{S}_3$.

Thus, $\mathcal{ALC}$ is at least as expressive as one-way $\mathcal{E}$-connections. Moreover, the syntactic fragment of $\mathcal{ALC}$ defined by $.^\sharp$ is expressively complete over bridged models for one-way $\mathcal{E}$-connections based on $\mathcal{ALC}$ components.

These simple reductions are not possible, however, if more expressive $\mathcal{E}$-connections are considered, for instance when using Booleans on links, or link constraints. In the next section, we give some concrete examples of the expressive limitations of basic $\mathcal{E}$-connections.

2.2 Expressivity of $\mathcal{E}$-Connections

It is sometimes claimed that one MOL $\mathcal{L}_1$ is more expressive than some other MOL $\mathcal{L}_2$. However, these claims are more often supported by reference to 'intuitive' arguments than by formal proofs. First, in order to compare the expressivity of two languages, they need to be interpreted in the same models, or in classes of models that are 'sufficiently similar'.

We first study the expressivity of $\mathcal{E}$-connections, and later compare it to DDLs.

Given that the basic $\mathcal{E}$-connection of any finite number of decidable DLs (expressible as ADSs) is decidable as well, it is clear that the interaction between the components has to be rather limited. Yet, it is not obvious what exactly can and what cannot be expressed in the combined language.

Suppose we construct a knowledge base containing information about relationships between people, companies, countries in the EU, etc., and suppose this KB is based on an $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{SHIQ}, \mathcal{ALCO})$. We use $\mathcal{SHIQ}$ to represent knowledge about people etc., $\mathcal{ALCO}$ to talk about countries, and use link relations being interpreted as, e.g., 'has citizenship in'.

Suppose we want to extend such a knowledge base with the following information:[5]

1. 'Children have the citizenship of their parents';
2. 'If a company cooperates with another company then the countries from which they operate have diplomatic relations'.

Assume we have link relations C for 'having citizenship in' and O for 'operating from', as well as roles has_child and cooperate of $\mathcal{SHIQ}$, and a role diplomatic of $\mathcal{ALCO}$. Then these constraints can easily be expressed in the language of first-order logic, compare Figure 1, as:

(†) $\forall x \forall y \forall z\ \big((x \text{ has\_child } y \wedge x\, C\, z) \rightarrow y\, C\, z)\big);$

(‡) $\forall x \forall y \forall x' \forall y' \big((x \text{ cooperate } y \wedge x\, O\, x' \wedge y\, O\, y') \rightarrow x' \text{ diplomatic } y'\big).$

Unfortunately, the $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{SHIQ}^\sharp, \mathcal{ALCO}^\sharp)$ is not expressive enough to enforce these conditions. To make this precise, we will work with the following definition of **definability** in $\mathcal{E}$-connections, where by **property** we shall mean any condition specified in the $\mathcal{E}$-connections analogue of a first-order correspondence language, compare [Kutz, 2004].

[5]These constraints are obviously not true generally. For instance, (1) is false according to US law, but true according to German law. Nevertheless, they can serve as prescriptions or regulations, e.g., of the new EU constitution.

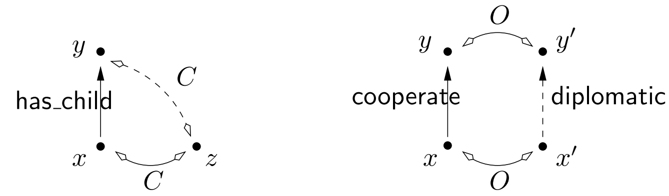

Figure 1: Undefinable Properties

Definition 6 [Definability in $\mathcal{E}$-Connections] *Let $\mathcal{C}$ be an $\mathcal{E}$-connection. A property $\mathcal{P}$ of models of $\mathcal{C}$ is called **definable in** $\mathcal{C}$ if there exists a finite set Γ of assertions of $\mathcal{C}$ such that, for all models $\mathfrak{M}$ of $\mathcal{C}$, the following holds:*

$$\mathfrak{M} \text{ has } \mathcal{P} \iff \mathfrak{M} \models \Gamma.$$

As is well known, undefinability results in modal logic—such as the undefinability of the irreflexivity of a Kripke frame—are usually gained by the concept of **bisimulation**. Similarly, the undefinability of properties (†) and (‡) in basic $\mathcal{E}$-connections can be shown by first lifting the concept of bisimulation to cover DLs, and then by generalising bisimulations to $\mathcal{E}$-connections.

Theorem 7 [Undefinability] *(†) and (‡) are not definable in basic $\mathcal{E}$-connections.*

However, this lack of expressivity is overcome by $\mathcal{E}$-connections that allow for so called *link constraints*. Clearly, we can simply add these kinds of constraints as new *primitive* assertions to $\mathcal{E}$-connections, obtaining various ways of increasing the expressive power of $\mathcal{E}$-connections. Thus, it is an interesting question to find out what kinds of first-order constraints are 'harmless' from the computational point of view.

A general investigation of this question seems to be rather complex. However, the constraints of the form above have the same structure in the sense that they enforce a new E-link between the models under certain conditions. In description logics, constraints of this form do not harm the transfer of decidability, compare [Kutz *et al.*, 2004].

We now introduce (basic) DDLs, reduce reasoning in DDLs to one-way $\mathcal{E}$-connections, and show that basic DDLs are strictly less expressive than a very weak fragment of basic $\mathcal{E}$-connections that we call negation-free unarily linked one-way $\mathcal{E}$-connections.

3 Distributed Description Logics (DDL)

Distributed description logics (DDLs) were originally proposed by Borgida and Serafini [Borgida and Serafini, 2002], and further studied in [Borgida and Serafini, 2003].

We start with a brief, but self-contained, description of the DDL formalism. Suppose that 2 description logics $\mathcal{S}_1$ and $\mathcal{S}_2$ are given. A pair $\mathfrak{D} = (\mathcal{S}_1, \mathcal{S}_2)$ is then called a **distributed description logic (DDL)**. We use subscripts to indicate that some concept C_i belongs to the language of the description logic $\mathcal{S}_i$. Two types of assertions—bridge rules and individual correspondence—are used to establish interconnections between the components of a DDL.

Definition 8 [Bridge Rules] *Let C_i and C_j be concepts from $\mathcal{S}_i$ and $\mathcal{S}_j$ ($i \neq j$), respectively. A* ***bridge rule*** *is an expression of the form*

$$C_i \xrightarrow{\sqsubseteq} C_j \qquad \text{(into rule)}$$

or of the form

$$C_i \xrightarrow{\sqsupseteq} C_j. \qquad \text{(onto rule)}$$

Let a_i be an object name of $\mathcal{S}_i$ and $b_j, b_j^1, \dots, b_j^n$ object names of $\mathcal{S}_j$. A ***partial individual correspondence*** *is an expression of the form*

$$a_i \mapsto b_j. \qquad \text{(PIC)}$$

A ***complete individual correspondence*** *is an expression of the form*

$$a_i \overset{=}{\mapsto} \{b_j^1, \dots, b_j^n\}. \qquad \text{(CIC)}$$

A ***distributed TBox*** *$\mathfrak{T}$ consists of TBoxes T_i of $\mathcal{S}_i$ together with a set of bridge rules. A* ***distributed ABox*** *$\mathfrak{A}$ consists of ABoxes A_i of $\mathcal{S}_i$ together with a set of partial and complete individual correspondences. A* ***distributed knowledge base*** *is a pair $(\mathfrak{T}, \mathfrak{A})$.*

The semantics of distributed knowledge bases is defined as follows.

Definition 9 [Semantics of DDL] *A* ***distributed interpretation*** *$\mathfrak{I}$ of a distributed knowledge base $(\mathfrak{T}, \mathfrak{A})$ as above is a pair $(\{\mathfrak{I}_i\}_{i\leq n}, \mathcal{R})$, where each $\mathfrak{I}_i$ is a model for the corresponding $\mathcal{S}_i$ and $\mathcal{R}$ is a function associating with every pair (i,j), $i \neq j$, a binary relation $r_{ij} \subseteq W_i \times W_j$ between the domains W_i and W_j of $\mathfrak{I}_i$ and $\mathfrak{I}_j$, respectively. Given a point $u \in W_i$ and a subset $U \subseteq W_i$, we set*

$$r_{ij}(u) = \{v \in W_j \mid (u,v) \in r_{ij}\}, \qquad r_{ij}(U) = \bigcup_{u\in U} r_{ij}(u).$$

The ***truth-relation*** *is standard for formulae of the component DLs. For bridge rules and individual correspondences it is defined as follows:*

$$\begin{array}{rcl} \mathfrak{I} \models C_i \xrightarrow{\sqsubseteq} C_j & \text{iff} & r_{ij}(C_i^{\mathfrak{I}}) \subseteq C_j^{\mathfrak{I}} \\ \mathfrak{I} \models C_i \xrightarrow{\sqsupseteq} C_j & \text{iff} & r_{ij}(C_i^{\mathfrak{I}}) \supseteq C_j^{\mathfrak{I}} \\ \mathfrak{I} \models a_i \mapsto b_j & \text{iff} & b_j^{\mathfrak{I}} \in r_{ij}(a_i^{\mathfrak{I}}) \\ \mathfrak{I} \models a_i \overset{=}{\mapsto} \{b_j^1, \dots, b_j^n\} & \text{iff} & r_{ij}(a_i^{\mathfrak{I}}) = \{(b_j^1)^{\mathfrak{I}}, \dots, (b_j^n)^{\mathfrak{I}}\} \end{array}$$

As usual, $\mathfrak{T} \models C \sqsubseteq D$ means that for every distributed interpretation $\mathfrak{I}$, if $\mathfrak{I} \models \varphi$ for all $\varphi \in \mathfrak{T}$, then $\mathfrak{I} \models C \sqsubseteq D$. The same definition applies to ABoxes $\mathfrak{A}$ and individual assertions.

It is of interest to note that, unlike $\mathcal{E}$-connections, DDLs do not provide new concept-formation operators to link the components of the DDL: both bridge rules and individual correspondences are assertions, and so atoms of knowledge bases, but not part of the concept language.

The satisfiability problem for distributed knowledge bases without complete individual correspondences (CIC) is easily reduced to the satisfiability problem for unarily linked one-way $\mathcal{E}$-connections.[6]

[6] The n-ary case is treated in [Kutz, 2004].

3.1 Reduction of DDLs to one-way $\mathcal{E}$-Connections

Fix a DDL $\mathfrak{D} = (\mathcal{S}_1, \mathcal{S}_2)$ and associate with it the $\mathcal{E}$-connection $\mathfrak{D}^{\mathcal{E}} = \mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$, where $\mathcal{E} = \{E_{12}, E_{21}\}$ consists of 2 binary relations. This is a unarily linked one-way $\mathcal{E}$-connection.

Definition 10 [Translation] *Suppose that $\mathfrak{K} = (\mathfrak{T}, \mathfrak{A})$ is a distributed knowledge base for $\mathfrak{D} = (\mathcal{S}_1, \mathcal{S}_2)_{i\leq n}$ without complete individual correspondences. We define a translation $.^\star$ from $\mathfrak{D}$-assertions to $\mathfrak{D}^{\mathcal{E}}$-assertions as follows: if φ is neither a bridge rule nor an individual correspondence then put $\varphi^\star = \varphi$; otherwise*

$$\begin{array}{rcl} (C_i \xrightarrow{\sqsubseteq} C_j)^\star & = & \langle E^{ij} \rangle C_i^\star \sqsubseteq C_j^\star; \\ (C_i \xrightarrow{\sqsupseteq} C_j)^\star & = & \langle E^{ij} \rangle C_i^\star \sqsupseteq C_j^\star; \\ (a_i \mapsto a_j)^\star & = & a_i E_{ij} a_j. \end{array}$$

Finally, we put $\mathfrak{T}^\star = \{\varphi^\star \mid \varphi \in \mathfrak{T}\}$, $\mathfrak{A}^\star = \{\varphi^\star \mid \varphi \in \mathfrak{A}\}$ and $\mathfrak{K}^\star = \mathfrak{T}^\star \cup \mathfrak{A}^\star$.

Note that we only need simple link assertions to translate partial individual correspondences: no application of link operators to object names is required. The theorem below follows now easily from the definition of the translation $\cdot^\star$:

Theorem 11 [Reduction] *A distributed knowledge base $\mathfrak{K}$ for a DDL $\mathfrak{D}$ without CICs is satisfiable if and only if $\mathfrak{K}^\star$ is satisfiable in a model of the one-way $\mathcal{E}$-connection $\mathfrak{D}^{\mathcal{E}}$.*

As a corollary, the satisfiability problem for DDLs $(\mathcal{S}_1, \mathcal{S}_2)$ without complete individual correspondences is decidable whenever the satisfiability problem for ABoxes relative to TBoxes is decidable for each of the $\mathcal{S}_i$. Unfortunately, complete individual correspondences cannot be translated into basic $\mathcal{E}$-connections, and, in fact, decidability transfer does not hold for arbitrary distributed description logics with knowledge bases including complete individual correspondences. However, it does hold when the component DLs satisfy a condition called *number tolerance*, which holds, for instance, if all DLs are fragments of $\mathcal{SHOIQ}$ without nominals. These results are shown in [Kutz *et al.*, 2004].

3.2 Expressivity of DDLs

The reduction of $\mathcal{E}$-connections of DLs and DDLs to one of the component DLs shows that these formalism do not add extra expressive power as compared to the most expressive component DL.[7] However, they impose a syntactic discipline on the use of link relations. In $\mathcal{E}$-connections, link relations can be used to build new complex concepts, while in DDLs they are exclusively used in bridge rules. We here study the effects of enriching the concept language as opposed to adding bridge rules as in DDL. We do this by looking at the simplest problem, the definability problem on link relations only.

Definition 12 *Let $\mathfrak{X}$ be a distributed knowledge base $(\mathfrak{T}, \mathfrak{A})$ of DDL $\mathfrak{D} = (\mathcal{S}_1, \mathcal{S}_2)$, or a knowledge base Γ of an $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$, sharing the link relations in $\mathcal{E} = (E_j)_{j\in J}$. Let ϕ be a first-order formula in the language $(E_j)_{j\in J}$. We say that $\mathfrak{X}$* ***defines*** *ϕ if $\mathcal{E}^{\mathfrak{M}} \models \phi$ for all $\mathfrak{M} \models \mathfrak{X}$.*

[7] As long as the target DL supports the translations $.^\sharp$ and $.^\star$.

Note that universal link operators of the form $[E_{ij}]C := \neg \langle E_{ij} \rangle \neg C$ are not available via bridge rules. In fact, not even negation on bridge rules is introduced in [Borgida and Serafini, 2003]. To bring DDLs and $\mathcal{E}$-connections even closer together, let us introduce the **negation-free** fragment of unarily linked one-way $\mathcal{E}$-connections by disallowing negation to be used on the primitive existential link operators.

The following gives an example of a simple property of links that can be defined already in the negation-free fragment of unarily linked one-way $\mathcal{E}$-connections.

Example 13 [Domain-Range Disjointness] *Consider the 1-assertion ϕ of a one-way $\mathcal{E}$-connection*

$$\phi := \langle E_{12} \rangle \langle E_{21} \rangle \top_1 \sqsubseteq \bot_1.$$

We claim that ϕ defines the property

$$(dis_1) \quad \forall x \forall y \forall z. x E_{12} y \to \neg y E_{21} z,$$

i.e., the range of E_{12} does not overlap the domain of E_{21}: $range(E_{12}) \cap dom(E_{21}) = \emptyset$. To obtain only non-trivial models, the link relations may be instantiated by adding link assertions $aE_{12}c$ and $bE_{21}d$.

Negation-free unarily-linked one-way $\mathcal{E}$-connections are the simplest variant of $\mathcal{E}$-connections we can think of, and are as close to DDLs as one can get. However:

Theorem 14 *Negation-free unarily linked one-way $\mathcal{E}$-connections are strictly more expressive than basic DDLs.*

We now turn our attention to P-DL. We show that P-DL imports statements can be omitted from the syntax, and reduce the P-DL satisfiability problem to a non-standard reasoning problem called **S**-compatibility.

4 Package-based Description Logics (P-DL)

We start with a brief, but self-contained, description of the P-DL formalism. Suppose that $\mathcal{S}$ is a description logic. A TBox $\mathcal{T}$ expressed in $\mathcal{S}$ together with a distinguished subset $\mathbf{S}$ of its signature $\mathsf{Sig}(\mathcal{T})$ constitute a *package*. Intuitively, the signature $\mathbf{S}$ contains the symbols whose meaning is defined 'within' $\mathcal{T}$, whereas the meaning of the remaining symbols in $\mathsf{Sig}(\mathcal{T})$ is 'borrowed' from other packages. More precisely, a package is formally defined as follows:

Definition 15 *A **package** $\mathcal{P} = (\mathsf{Ho}(\mathcal{P}), \mathcal{T}(\mathcal{P}))$ is given by a TBox $\mathcal{T}(\mathcal{P})$ and a signature $\mathsf{Ho}(\mathcal{P}) \subseteq \mathsf{Sig}(\mathcal{T}(\mathcal{P}))$. If a symbol t is in $\mathsf{Ho}(P)$, we say that $\mathcal{P}$ is the **home package** of t. If C is a (possibly complex) concept such that C contains at least one symbol in $\mathsf{Ho}(\mathcal{P})$, we say that C is a $\mathcal{P}$**-concept**.*

Suppose we want to integrate a set of packages $\mathcal{P}_1, \ldots, \mathcal{P}_n$, expressed in $\mathcal{S}$. Intuitively, each package $\mathcal{P}_i$ defines the meaning of symbols of $\mathsf{Ho}(\mathcal{P}_i)$ in $\mathcal{T}(\mathcal{P}_i)$, and may 'borrow' the meaning of other symbols in $\mathsf{Sig}(\mathcal{T}(\mathcal{P}_i)) \setminus \mathsf{Ho}(\mathcal{P}_i)$ from the remaining packages $\mathcal{P}_j, j \neq i$. We shall assume that each symbol occurring in $\mathcal{P}_1, \ldots, \mathcal{P}_n$ has a (single) home package. Hence, given $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and a symbol s, there should be no ambiguity in determining the home package of s. This intuition leads to the notion of a *package-based ontology*. We first define *imports statements* that are used to 'borrow' the meaning of symbols from foreign packages.

Definition 16 *Given two packages $\mathcal{P}_i, \mathcal{P}_j$, $i \neq j$, and a symbol $s \in \mathsf{Ho}(\mathcal{P}_j)$, an **import statement** for $\mathcal{P}_i$ is an expression of the form $\mathcal{P}_j \xrightarrow{s} \mathcal{P}_i$. We say that $\mathcal{P}_i$ **imports** s from $\mathcal{P}_j$.*

We are now ready to define a package-based ontology.

Definition 17 *A pair $\mathcal{O} = (\mathfrak{P}, \mathfrak{I})$ is a **package-based ontology** (pb-ontology) if $\mathfrak{P} = (\mathcal{P}_i)_{i \leq n}$ is an n-tuple of packages such that $\mathsf{Ho}(\mathcal{P}_i) \cap \mathsf{Ho}(\mathcal{P}_j) = \emptyset$, for $i \neq j$ and $n \in \mathbb{N}$, and $\mathfrak{I} = \{\alpha_k\}_{k \leq m}$ is a finite set of import statements for packages from $\mathfrak{P}$. We say that $\mathcal{O}$ has **width** n.*

The semantics of pb-ontologies is as follows:

Definition 18 [Semantics] *A (distributed) interpretation $\Im$ for a pb-ontology $\mathcal{O} = (\mathfrak{P}, \mathfrak{I})$ of width n is a pair $\Im = ((\mathcal{I}_i)_{i \leq n}, \mathcal{R})$, where $\mathcal{I}_i = (\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i})$ is an interpretation for $\mathsf{Sig}(\mathcal{T}(\mathcal{P}_i))$ and $\mathcal{R}$ is a function associating with each pair $(i, j), i \neq j$, a bijection $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ defined on a subset of $\Delta^{\mathcal{I}_i}$, and such that, for all i, j, k, we have $r_{ij} \circ r_{jk} = r_{ik}$ and $|\mathsf{dom}(r_{jk})| = |\mathsf{range}(r_{ij})|$.*

*An interpretation $\Im$ **satisfies** the package $\mathcal{P}_i$ if $\mathcal{I}_i \models \mathcal{T}_i$, and satisfies the import statement $\mathcal{P}_j \xrightarrow{s} \mathcal{P}_i$ if $r_{ij}(s^{\mathcal{I}_i}) = s^{\mathcal{I}_j}$. The interpretation $\Im$ satisfies $\mathcal{O}$ if $\Im$ satisfies every package and every import statement in $\mathcal{O}$. In this case, we say $\Im$ is a **model** of $\mathcal{O}$, written $\Im \models \mathcal{O}$. A pb-ontology $\mathcal{O}$ is **satisfiable** if it has a model. A $\mathcal{P}_i$-concept C is satisfiable w.r.t. $\mathcal{O}$ if there is a model of $\mathcal{O}$ in which $C^{\mathcal{I}_i} \neq \emptyset$.*

Intuitively, the semantics assigns a 'local' model to each package. When a package $\mathcal{P}_i$ imports a symbol s in the home signature of another package $\mathcal{P}_j$, the local models $\mathcal{I}_i, \mathcal{I}_j$ must agree on the meaning of the imported symbols. Such an agreement is formally given by a bijection r_{ij} from (a subset of) the domain $\Delta^{\mathcal{I}_i}$ onto (a subset of) $\Delta^{\mathcal{I}_j}$. Corresponding elements in $\Delta^{\mathcal{I}_i}$ and $\Delta^{\mathcal{I}_j}$ are instances of the same shared concepts and are connected by the (interpretation of) the same shared roles. Moreover, by requiring that those bijections compose, it is ensured that such an agreement can be reached across several packages.

4.1 Imports-free P-DL

In this section, we provide a simpler, yet equivalent, formulation of the P-DL formalism that we call *imports-free* P-DL. The goal of this simplification is, on the one hand, to remove the imports statements from the syntax and, on the other hand, to replace the bijections r_{ij} in the semantics by actual identities between objects in different local interpretations.

We start by noting that import statements are non-directional:

Proposition 19 [Non-directionality] *Let $\Im$ be an interpretation for a pb-ontology $\mathcal{O}$ and $\mathcal{P}_i \xrightarrow{s} \mathcal{P}_j$ an import statement.*

$$\Im \models \mathcal{P}_i \xrightarrow{s} \mathcal{P}_j \quad \iff \quad \Im \models \mathcal{P}_j \xrightarrow{s} \mathcal{P}_i$$

Based on this observation, define, for a given pb-ontology $\mathcal{O} = (\mathfrak{P}, \mathfrak{I})$ of width n, the set of symbols that occur in import statements in $\mathcal{O}$ as

$$\mathsf{Imp}(\mathcal{O}) := \{s \mid \mathcal{P}_i \xrightarrow{s} \mathcal{P}_j \in \mathfrak{I}\}.$$

Next, we note that, according to Definition 17, a symbol s may occur in packages $\mathcal{P}_i$ and $\mathcal{P}_j$ without being imported by either. Intuitively, s behaves as a different name in $\mathcal{P}_i$ and $\mathcal{P}_j$ in the sense that the interpretation of s in $\mathcal{P}_i$ does not influence its interpretation in $\mathcal{P}_j$, and vice-versa. We can formalise this observation by showing that P-DLs are invariant under transformations that 'rename' the shared symbols that do not belong to the imported signature.

Definition 20 *Let $\mathbf{S}, \mathbf{S}'$ be disjoint signatures of the same size. A* ***substitution*** *is a bijection $\mathfrak{s} : \mathbf{S} \to \mathbf{S}'$. Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we define $\mathcal{I}^{\mathfrak{s}} = (\Delta^{\mathcal{I}^{\mathfrak{s}}}, \cdot^{\mathcal{I}^{\mathfrak{s}}})$ as follows: (1) $\Delta^{\mathcal{I}^{\mathfrak{s}}} = \Delta^{\mathcal{I}}$; (2) for every $t \notin \mathbf{S}$: $t^{\mathcal{I}^{\mathfrak{s}}} = t^{\mathcal{I}}$; (3) for every $s \in \mathbf{S}$: $\mathfrak{s}(s)^{\mathcal{I}^{\mathfrak{s}}} = s^{\mathcal{I}}$.*

The invariance result is provided by the following proposition, where we restrict our attention for simplicity and w.l.o.g. to pb-ontologies of width 2:

Proposition 21 [Substitution Invariance] *Let $\mathcal{O}$ be a pb-ontology $(\langle \mathcal{P}_1, \mathcal{P}_2\rangle, \mathfrak{I})$ of width 2 with $\mathcal{P}_i = (\mathsf{Ho}(\mathcal{P}_i), \mathcal{T}_i)$, $i = 1, 2$. Let $\mathbf{S} = (\mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)) \setminus \mathsf{Imp}(\mathcal{O})$, $\mathbf{S}'$ a signature disjoint with $\mathsf{Sig}(\mathcal{O})$ and $|\mathbf{S}| = |\mathbf{S}'|$, and let $\mathfrak{s} : \mathbf{S} \to \mathbf{S}'$ be a substitution. Then for every interpretation $\mathfrak{I} = (\langle \mathcal{I}_1, \mathcal{I}_2\rangle, \mathcal{R})$:*

$$\mathfrak{I} \models \mathcal{O} \iff (\langle \mathcal{I}_1, \mathcal{I}_2^{\mathfrak{s}}\rangle, \mathcal{R}) \models (\langle \mathcal{P}_1, \mathcal{P}_2^{\mathfrak{s}}\rangle, \mathfrak{I})$$

The proposition above implies that, without loss of generality, we can assume that for any pair of packages $\mathcal{P}_i, \mathcal{P}_j$ we have $\mathsf{Sig}(\mathcal{T}_i) \cap \mathsf{Sig}(\mathcal{T}_j) \subseteq \mathsf{Imp}(\mathcal{O})$. Thus, the imports statements can be already uniquely identified by the the set $\mathsf{Imp}(\mathcal{O})$ of an adequately pre-processed (but equivalent) pb-ontology where non-imported symbols have been renamed, and can hence be suppressed from the syntax.

Finally, we observe that the bijection r_{ij} in the semantics of P-DL can be equivalently represented by just copying isomorphically the relevant partial domain from $\mathcal{I}_j$ into $\mathcal{I}_i$ such that the local models 'agree' on the shared symbols. Simply note that the imports statements just indicate which (two) local interpretations should agree on the interpretation of a given symbol, and these symbols occur in both packages involved in the imports statement.

These observations lead to a simpler, yet equivalent, formulation of P-DL:

Definition 22 *An* ***imports-free*** *packaged-based ontology (if-ontology) is a tuple $\mathcal{O} = (\mathcal{T}_i)_{i \leq n}$ of TBoxes.*

The semantics of imports-free P-DL is given as follows:

Definition 23 [Semantics] *Let $\mathcal{O} = (\mathcal{T}_i)_{i \leq n}$ be an if-ontology. A (distributed) interpretation $\mathfrak{I}$ for $\mathcal{O}$ is a tuple $\mathfrak{I} = (\mathcal{I}_i)_{i \leq n}$ of interpretations $\mathcal{I}_i = (\Delta_i, \cdot^{\mathcal{I}_i})$ for $\mathsf{Sig}(\mathcal{T}_i)$.*

The interpretation $\mathfrak{I}$ ***satisfies*** *the TBox $\mathcal{T}_i$ iff $\mathcal{I}_i \models \mathcal{T}_i$, and it satisfies $\mathcal{O}$ iff it satisfies $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and for every symbol s in $\mathsf{Sig}(\mathcal{T}_i) \cap \mathsf{Sig}(\mathcal{T}_j)$, for $1 \leq i, j \leq n$, we have $s^{\mathcal{I}_i} = s^{\mathcal{I}_j}$. In this case $\mathfrak{I}$ is a* ***model*** *of $\mathcal{O}$, written $\mathfrak{I} \models \mathcal{O}$. An if-ontology $\mathcal{O}$ is* ***satisfiable*** *iff it has a model.*

The following proposition shows that both formalisms can be seen as equivalent:

Proposition 24 *Let $\mathcal{O} = \langle (\mathcal{P}_i)_{i \leq n}, \mathfrak{I}\rangle$ be a pb-ontology such that $\mathsf{Sig}(\mathcal{T}_i) \cap \mathsf{Sig}(\mathcal{T}_j) \subseteq \mathsf{Imp}(\mathcal{O})$, for $i \neq j$. Let $\mathcal{O}' = (\mathcal{T}_i)_{i \leq n}$ be the if-ontology composed of the TBoxes $\mathcal{T}_i$ in $\mathcal{P}_i$. Let $\mathfrak{I} = ((\mathcal{I}_i)_{i \leq n}, \mathcal{R})$ be an interpretation for $\mathcal{O}$ and $\mathfrak{I}' = (\mathcal{I}_i)_{i \leq n}$ be the interpretation for $\mathcal{O}'$ containing the same local models.*

$$\mathfrak{I} \models \mathcal{O} \iff \mathfrak{I}' \models \mathcal{O}'$$

Consequently, in what follows and without loss of generality, we will focus on imports-free P-DL.

4.2 From P-DL Satisfiability to TBox consistency

In this section, we present a new reasoning problem for description logics, that we call *compatibility* of a set of TBoxes $\mathcal{T}_1, \ldots, \mathcal{T}_n$ w.r.t. a signature $\mathbf{S}$, or $\mathbf{S}$-compatibility. We show that the satisfiability problem for P-DL and the $\mathbf{S}$-compatibility problem are inter-reducible for any description logic $\mathcal{S}$. This result implies that, for any description logic, we can simply focus, without loss of generality, on the $\mathbf{S}$-compatibility problem. In the case of the description logic $\mathcal{SHIQ}$, we additionally show that the $\mathbf{S}$-compatibility problem and the standard TBox consistency problem are also inter-reducible. Finally, in the case of $\mathcal{SHOIQ}$, we argue that this is not the case, and hypothesise that the $\mathbf{S}$-compatibility problem is indeed harder than the standard TBox consistency problem.

The S-Compatibility Problem

In this section, we define the compatibility problem for a set $\mathcal{T}_1, \ldots, \mathcal{T}_n$ of TBoxes w.r.t. a signature $\mathbf{S}$. For simplicity in the presentation and without loss of generality, we will restrict ourselves to the case in which $n = 2$.

Intuitively, given the TBoxes $\mathcal{T}_1, \mathcal{T}_2$ expressed in a description logic $\mathcal{S}$ and a signature $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)$, we will say that they are $\mathbf{S}$-*compatible* if we can find an interpretation for the symbols in $\mathbf{S}$ that can be *extended* to both a model to $\mathcal{T}_1$ and a model of $\mathcal{T}_2$ by interpreting the additional predicates and possibly adding new elements to the interpretation domain. Obviously, these models agree on the interpretation of the symbols in $\mathbf{S}$. We first define the notion of an *expansion* of an interpretation.

Definition 25 [Expansion] *An $\mathbf{S}$-interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is an* ***expansion*** *of an $\mathbf{S}'$-interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ if (1) $\mathbf{S} \supseteq \mathbf{S}'$, (2) $\Delta^{\mathcal{I}} \supseteq \Delta^{\mathcal{I}'}$, and (3) $s^{\mathcal{I}} = s^{\mathcal{I}'}$ for every $s \in \mathbf{S}'$.*

Note that the interpretation $\mathcal{I}$:

- coincides with $\mathcal{I}'$ in the interpretation of symbols in $\mathbf{S}'$,
- 'freely' interprets the symbols in $\mathbf{S} \setminus \mathbf{S}'$, and
- possibly expands the interpretation domain of $\mathcal{I}'$.

We are now ready to define the compatibility problem:

Definition 26 [S-compatibility] *Let $\mathcal{T}_1, \mathcal{T}_2$ be TBoxes expressed in a description logic $\mathcal{S}$, and let $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)$. We say that $\mathcal{T}_1, \mathcal{T}_2$ are* ***$\mathbf{S}$-compatible*** *if there exists an interpretation $\mathcal{I}_{\mathbf{S}}$ of the symbols in $\mathbf{S}$ that can be expanded to a model $\mathcal{I}_1$ of $\mathcal{T}_1$ and to a model $\mathcal{I}_2$ of $\mathcal{T}_2$. In this case, we say that $\mathcal{I}_1, \mathcal{I}_2$ are* ***$\mathbf{S}$-compatible models****.*

P-DL Satisfiability and S-Compatibility

The notion of $\mathbf{S}$-compatibility is especially relevant for understanding the nature of P-DL. In particular, the following proposition shows that, for any description logic $\mathcal{S}$, the satisfiability problem for if-ontologies, and therefore also for pb-ontologies, and the $\mathbf{S}$-compatibility problem can be reduced to each other.

Proposition 27 [P-DL Reduction] *Let $\mathcal{O} = \langle \mathcal{T}_1, \mathcal{T}_2\rangle$ be an if-ontology. Then $\mathcal{O}$ is satisfiable if and only if $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{S}$-compatible for $\mathbf{S} = \mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)$.*

In what follows, we will restrict ourselves to investigating the $\mathbf{S}$-compatibility problem for various DLs.

$\mathbf{S}$-Compatibility and TBox Consistency for $\mathcal{SHIQ}$
Suppose ontologies $\mathcal{T}_1, \mathcal{T}_2$ are expressed in $\mathcal{SHIQ}$. The following theorem shows that deciding $\mathbf{S}$-compatibility of $\mathcal{T}_1, \mathcal{T}_2$ is equivalent to deciding the (ordinary) ontology consistency problem for the merged ontology $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$.

Theorem 28 *The $\mathcal{SHIQ}$ TBoxes $\mathcal{T}_1, \mathcal{T}_2$ are $\mathbf{S}$-compatible, for $\mathbf{S} = \mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)$, if and only if the TBox $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ is consistent.*

The following corollary immediately follows from Theorem 28 and Proposition 27:

Corollary 29 *For any $\mathcal{SHIQ}$ TBoxes $\mathcal{T}_1, \mathcal{T}_2$, the* if*-ontology $\mathcal{O} = \langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ is satisfiable if and only if the TBox $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ is consistent.*

Corollary 29 contradicts the original intuition underlying the semantics of P-DL. In fact, localised semantics as described in [Bao *et al.*, 2006c; 2006b] are not supported by P-DL since the existence of a *global model* (i.e. a model of the union of the packages) is indeed required. The corollary also implies that, in order to reason about an if-ontology $\mathcal{O} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$, it suffices to perform standard DL reasoning over the ontology $\mathcal{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_n$. No specialised tableau algorithm, such as the one presented in [Bao *et al.*, 2006a] for $\mathcal{ALC}$, is indeed required.

The $\mathbf{S}$-compatibility Problem for $\mathcal{SHOIQ}$
Contrary to the case of $\mathcal{SHIQ}$, the compatibility problem for a tuple of $\mathcal{SHOIQ}$ Tboxes is not reducible to the consistency problem of the union of the TBoxes:

Proposition 30 *There exist $\mathbf{S}$-compatible $\mathcal{SHOIQ}$ TBoxes $\mathcal{T}_1, \mathcal{T}_2$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{T}_1) \cap \mathsf{Sig}(\mathcal{T}_2)$ such that $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ is inconsistent.*

For a proof, simply consider the following TBoxes:

$$\begin{aligned} \mathcal{T}_1 &= \{\top \sqsubseteq i \sqcup j\} \\ \mathcal{T}_2 &= \{\top \sqsubseteq i\} \end{aligned}$$

and the signature $\mathbf{S} = \{i\}$. The TBox $\mathcal{T}_1$ only has models of cardinality 2, whereas $\mathcal{T}_2$ has only models of cardinality 1. Obviously, $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ is inconsistent. However, $\mathcal{T}_1$ and $\mathcal{T}_2$ are still $\mathbf{S}$-compatible.

Intuitively, the proof of Theorem 28 relies on the fact that, for $\mathcal{SHIQ}$, whenever $\mathcal{T}_1, \mathcal{T}_2$ have a pair of $\mathbf{S}$-compatible models, then they also have a pair of *countably infinite* $\mathbf{S}$-compatible models, which is no longer the case for $\mathcal{SHOIQ}$. Obviously, if $\mathcal{T}_1 \cup \mathcal{T}_2$ is consistent, then $\mathcal{T}_1, \mathcal{T}_2$ are $\mathbf{S}$-compatible; however, as seen above, if $\mathcal{T}_1 \cup \mathcal{T}_2$ is inconsistent, it does not necessarily mean that $\mathcal{T}_1, \mathcal{T}_2$ are *not* $\mathbf{S}$-compatible.

This suggests that $\mathbf{S}$-compatibility for $\mathcal{SHOIQ}$ is harder than TBox satisfiability, yet likely to be decidable.[8] The investigation of the complexity of this problem is left for future work. The reader familiar with P-DL will note that the decidability of the consistency problem for pb-ontologies expressed in $\mathcal{SHOIQ}$ has been claimed in [Bao *et al.*, 2006c]. Unfortunately, the decidability proof given in [Bao *et al.*, 2006c] is incorrect. In particular, it does not provide a correct account for the example given after Proposition 30.

[8]We thank Prof Frank Wolter for valuable discussion about this particular issue.

5 Modularity without Modular Languages

It is a common belief that the use of a modular ontology language equipped with a non-standard semantics, such as $\mathcal{E}$-connections, DDL and P-DL, is crucial for establishing a notion of modularity in ontologies. In this section, we argue against this belief. In particular, we establish a set of reasoning services that can be used to formalise and solve the tasks presented in Section 1, and we show that the satisfiability problem in modular languages is closely related to these reasoning services.

It should come as no surprise that the (standard) satisfiability problem in a modular formalism (equipped with a non-standard semantics) can be formulated in terms of a non-standard reasoning service with standard semantics. In particular, we have shown in Section 4 that a quite compelling and rather sophisticated modular semantics, such as the one proposed in P-DL, can be equivalently captured by the $\mathbf{S}$-compatibility problem.

In this section, we first summarise the main intuitions underlying the design of modular ontology languages. We then show how these intuitions can be formalised as reasoning services. Since the connection between P-DL and reasoning services has already been established in Section 4, in what follows our discussion will be mainly focused on $\mathcal{E}$-connections and DDL.

5.1 Modular Ontology Languages in a Nutshell

In existing modular ontology languages, the signature of a module is partitioned into a 'local' signature and an 'external' signature. The former contains the symbols whose meaning is defined within the module; the latter contains the symbols that are defined externally in other modules. The local signature is described using a local language and its meaning is given by a local interpretation. The meaning of the external symbols is brought into the local context using different means, such as bridge rules (in the case of DDL), quantification over link relations (in the case of $\mathcal{E}$-Connections), and imports statements (in the case of P-DL). The semantics of DDL, $\mathcal{E}$-Connections and P-DL is tailored such that local and external symbols are treated differently.

Perhaps, the most fundamental intuitions behind the design of modular ontology formalisms are the following:

1. there should exist a notion of *localisation* of the knowledge within a module in order to limit and control the interaction with other modules. In particular, the semantics of negation and the semantics of GCIs in $\mathcal{E}$-connections, DDL and P-DL are *scoped* within the module in which they are represented, and
2. the use of the external symbols should *not* alter their meaning in their original context, i.e., the meaning of the external symbols is imported, but not changed. Thus, the formalism should provide *control* over the logical consequences concerning the external symbols.

Similar intuitions have been formalised as reasoning services in a series of recent papers [Cuenca-Grau *et al.*, 2007; Lutz *et al.*, 2007; Cuenca-Grau *et al.*, 2006c; Ghilardi *et al.*, 2006; Cuenca-Grau *et al.*, 2006a]. In [Cuenca-Grau *et al.*, 2006c; 2006a], the idea of knowledge localisation is formalised using the notion of *locality* of a TBox. Intuitively, a TBox is local if its axioms do not 'define' the top concept $\top$ and, thus, they do not constrain the meaning of 'everything in the world'. In [Ghilardi *et al.*, 2006; Lutz *et al.*, 2007; Cuenca-Grau *et al.*, 2006c], the notion of a *conservative extension* is used to formalise ontology refinements, safe mergings of ontologies, and extraction of modules. An ontology $\mathcal{T} \cup \mathcal{T}'$ is a conservative extension of an ontology $\mathcal{T}'$ w.r.t. the external signature $\mathbf{S}$ if every consequence of $\mathcal{T} \cup \mathcal{T}'$ formulated in $\mathbf{S}$ is already a consequence of $\mathcal{T}'$. Thus, the merge does not introduce new consequences concerning the external symbols.

In what follows, we formulate these reasoning services and investigate their relationship with the semantics and the satisfiability problem in modular ontology formalisms.

5.2 Locality

In this section, we introduce the notion of *locality* of a TBox. Local ontologies contain only GCIs with a limited 'global' effect and, in particular, do not fix the meaning of the universal concept $\top$. For example, assume that $\mathcal{T}_1$ is an ontology about vehicles and $\mathcal{T}_2$ an ontology about people. Suppose that $\mathcal{T}_1$ contains the axiom $\alpha = (\top \sqsubseteq \mathit{Vehicle})$ and $\mathcal{T}_2$ contains the axiom $\beta = (\neg \mathit{Man} \sqsubseteq \mathit{Woman})$. If $\mathcal{T}_1, \mathcal{T}_2$ are merged into $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, the axiom α forces every person to be a vehicle, whereas β forces every object that is not a man (such as a vehicle) to be a woman. The 'globality' of a GCI can be assessed using the notion of a *domain expansion* [Cuenca-Grau *et al.*, 2006c; 2006a].

Definition 31 [Domain Expansion] *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, .^{\mathcal{J}})$ be interpretations and ∇ a a non-empty set disjoint with $\Delta^{\mathcal{I}}$. We say that $\mathcal{J}$ is the **domain expansion** of $\mathcal{I}$ with ∇ if*

$$
\begin{aligned}
\Delta^{\mathcal{J}} &= \Delta^{\mathcal{I}} \cup \nabla \\
A^{\mathcal{J}} &= A^{\mathcal{I}} \quad \text{for each concept name} \\
R^{\mathcal{J}} &= R^{\mathcal{I}} \quad \text{for each role name}
\end{aligned}
$$

Intuitively, the interpretation $\mathcal{J}$ is identical to $\mathcal{I}$ except for the fact that it contains some additional elements in the interpretation domain. These elements do not participate in the interpretation of concepts or roles. The following question naturally arises: if $\mathcal{I}$ is a model of $\mathcal{T}$, is $\mathcal{J}$ also a model of $\mathcal{T}$? Local ontologies are precisely those whose models are closed under domain expansions [Cuenca-Grau *et al.*, 2006c; 2006a].

Definition 32 [Locality] *Let $\mathcal{T}$ be a TBox. We say that $\mathcal{T}$ is **local** if, for every $\mathcal{I} \models \mathcal{T}$ and every set ∇ disjoint with $\Delta^{\mathcal{I}}$, the expansion $\mathcal{J}$ of $\mathcal{I}$ with ∇ is a model of $\mathcal{T}$.*

The obvious question is how locality relates to the semantics of DDLs and $\mathcal{E}$-connections. In what follows, we show that the $\mathcal{E}$-connections and DDL semantics *force* each module to be local;

Proposition 33 [Locality and $\mathcal{E}$-connections] *Let Γ be a one-way $\mathcal{E}$-connection in the language $\mathcal{C}^{\mathcal{E}}(\mathcal{ALC}, \mathcal{ALC})$, then the $\mathcal{ALC}$ knowledge base $\Gamma^{\sharp}$ is local.*

For simplicity, we have considered only $\mathcal{ALC}$, but it is not hard to prove that this result also holds for basic $\mathcal{E}$-connections of $\mathcal{SHOIQ}$ ontologies (of course, with the appropriate translation function). In [Cuenca-Grau *et al.*, 2006c], it is shown that deciding locality of $\mathcal{SHOIQ}$ ontologies can be decided in polynomial time.

5.3 Conservative Extensions

One of the main reasons for using MOLs is to restrict the interaction between the (local) models of the modules in order to achieve a certain level of *control* over the consequences that knowledge in one module may have over the entailments in other modules.

A similar intuition has been recently formalised in the context of ontologies using the notion of a *conservative extension* [Ghilardi *et al.*, 2006; Lutz *et al.*, 2007]

Definition 34 [Conservative Extension] *Let $\mathcal{T}$ and $\mathcal{T}'$ be TBoxes formulated in $\mathcal{L}$ and $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{T}')$. Then $\mathcal{T} \cup \mathcal{T}'$ is an $\mathbf{S}$-**conservative extension** of $\mathcal{T}'$ if, for every $\mathcal{L}$-axiom α with $\mathsf{Sig}(\alpha) \subseteq \mathbf{S}$ we have $\mathcal{T} \cup \mathcal{T}' \models \alpha$ if and only if $\mathcal{T}' \models \alpha$.*

Deciding conservative extensions means to decide, given $\mathcal{T}$ and $\mathcal{T}'$ and a signature $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{T}')$, whether $\mathcal{T} \cup \mathcal{T}'$ is an $\mathbf{S}$-conservative extension of $\mathcal{T}'$.

Definition 34 states that, given $\mathcal{T}$ and $\mathcal{T}'$, their union $\mathcal{T} \cup \mathcal{T}'$ does not yield new consequences in $\mathcal{T}'$, provided that $\mathcal{T} \cup \mathcal{T}'$ is a conservative extension of $\mathcal{T}'$. This implies that the notion of a conservative extension provides a notion of 'control' over the logical consequences in the merge. The problem has been investigated in [Ghilardi *et al.*, 2006; Lutz *et al.*, 2007] and the first results concerning decidability and complexity have been established.

The obvious question is how the semantics of MOLs relate to conservative extensions. We consider here the case of $\mathcal{E}$-connections and prove that, in general, when combining two ontologies using one-way $\mathcal{E}$-connections, new consequences can appear in any of the component ontologies due to the influence of the other one.

Proposition 35 [$\mathcal{E}$-connections are not conservative] *There exists a one-way $\mathcal{E}$-connected KB $\Gamma = \langle \Gamma_1, \Gamma_2 \rangle$ without object assertions in the language $\mathcal{C}^{\mathcal{E}}(\mathcal{ALC}, \mathcal{ALC})$ and i-axioms α_i, for $i \in \{1, 2\}$, such that $\Gamma^{\mathcal{ALC}} \models \alpha_i^{\sharp}$ but $\Gamma_i^{\mathcal{ALC}} \not\models \alpha_i^{\sharp}$.*

The reader should not be surprised by such a result. In fact, in $\mathcal{E}$-connections, concepts in Γ_2 can be used to define concepts in Γ_1, and vice-versa, and hence mutual interaction between the connected ontologies is to be expected. However, in case one of the ontologies, say Γ_2, does not contain link relations pointing to Γ_1, we should expect Γ_2 to influence Γ_1, but not vice-verse. This is the case:

Proposition 36 *Let $\Gamma = \langle \Gamma_1, \Gamma_2 \rangle$ be a basic $\mathcal{E}$-connection in the language $\mathcal{C}^{\mathcal{E}}(\mathcal{ALC}, \mathcal{ALC})$ such that $\mathcal{E}_{21} = \emptyset$. Let α be a 2-axiom. Then*

$$\Gamma^{\mathcal{ALC}} \models \alpha^{\sharp} \iff \Gamma_2^{\mathcal{ALC}} \models \alpha^{\sharp}.$$

Intuitively, the proposition above implies that the different components of an $\mathcal{E}$-connected knowledge base interact only through the link relations and, hence, in order for the component Γ_2 to affect Γ_1, a link relation in $\mathcal{E}_{21}$ must exist. This is a consequence of the fact that $\mathcal{E}$-connection semantics localise the knowledge within each component and hence the effects of GCIs not mentioning the link relations explicitly do not propagate to other modules.

6 Conclusion

In this paper, we have investigated the relationship between various modular ontology languages proposed in the literature, discussed the appropriateness of their semantics to provide support for key tasks in ontology engineering, and, finally, we have established the relationship between these modular formalisms and various non-standard reasoning services recently proposed, such as locality and conservative extensions.

Our results raise the following question: should we aim at defining modular formalisms with non-standard semantics, or should we, on the contrary, aim at defining sensible *reasoning services* while keeping the conventional semantics unaltered?

The latter option seems to offer compelling advantages:

- The specification of ontology languages, such as OWL, does not need to be modified, which facilitates tool compatibility and tool implementation. Moreover, non-standard reasoning services can more easily be implemented on top of existing standard reasoning tasks.
- Modular semantics are likely to confuse the user; in particular, it is non-trivial to determine which formal properties should be expected from them. Furthermore, it is often hard to compare different modular semantics and to establish their relationship with the conventional semantics. An example is the case of P-DL for which we have shown that distributed reasoning over a set of packages is equivalent to the standard ontology consistency problem over the union of the packages for the logic $\mathcal{SHIQ}$.

However, there are clearly also arguments in favor of using modular ontology languages:

- Modular Languages provide a clean way of controlling the interaction between modules; in particular, we have shown that $\mathcal{E}$-connections provide locality (always) and conservative extensions (in some cases) ”for free”.
- It may be convenient from a modeling perspective to have special syntax in the ontology language for combining the different modules; in particular, it may be useful to distinguish explicitly which logical axioms “glue” the modules together. MOLs provide such a distinction in a clean way.

We believe that future research in the area should take into consideration these issues.

References

[Baader *et al.*, 2002] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research (JAIR)*, 16:1–58, 2002.

[Bao *et al.*, 2006a] J. Bao, D. Caragea, and V. Honavar. A Tableau-based Federated Reasoning Algorithm for Modular Ontologies. In *Proc. of the IEEE/WIC/ACM Int. Conf. on Web Intelligence*, 2006.

[Bao *et al.*, 2006b] J. Bao, D. Caragea, and V. Honavar. Modular Ontologies - A Formal Investigation of Semantics and Expressivity. In *Proc. of ASWC*. Springer, 2006.

[Bao *et al.*, 2006c] J. Bao, D. Caragea, and V. Honavar. On the Semantics of Linking and Importing in Modular Ontologies. In *Proc. of ISWC*. Springer, 2006.

[Borgida and Serafini, 2002] A. Borgida and L. Serafini. Distributed Description Logics: Directed Domain Correspondences in Federated Information Sources. In *On The Move to Meaningful Internet Systems*, volume 2519 of *LNCS*, pages 36–53. Springer, 2002.

[Borgida and Serafini, 2003] A. Borgida and L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics*, 1:153–184, 2003.

[Cuenca-Grau and Kutz, 2006] B. Cuenca-Grau and O. Kutz. Modular Ontology Languages Revisited. Technical report, The University of Manchester, `www.cs.man.ac.uk/~okutz/TRmol.pdf`, 2006.

[Cuenca-Grau *et al.*, 2006a] B. Cuenca-Grau, I. Horrocks, O. Kutz, and U. Sattler. Will my Ontologies Fit Together? In *Proc. of DL*, 2006.

[Cuenca-Grau *et al.*, 2006b] B. Cuenca-Grau, B. Parsia, and E. Sirin. Combining OWL Ontologies Using $\mathcal{E}$-Connections. *Journal of Web Semantics*, 4(1):40–59, 2006.

[Cuenca-Grau *et al.*, 2006c] B. Cuenca-Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and Web Ontologies. In *Proc. of KR*, 2006.

[Cuenca-Grau *et al.*, 2007] B. Cuenca-Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modular integration of ontologies. In *Proc. of IJCAI*, 2007. To Appear.

[Ghilardi *et al.*, 2006] S. Ghilardi, C. Lutz, and F. Wolter. Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In *Proc. of KR*, 2006.

[Kutz *et al.*, 2004] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-connections of Abstract Description Systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[Kutz, 2004] O. Kutz. *$\mathcal{E}$-Connections and Logics of Distance*. PhD thesis, The University of Liverpool, 2004.

[Lutz *et al.*, 2007] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of IJCAI*, 2007. To appear.