Automatic Partitioning of OWL Ontologies Using $\mathcal{E}$-Connections

Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, Aditya Kalyanpur
University of Maryland, College Park, MD, USA
bernardo@minswap.org;bparsia@isr.umd.edu;
{evren,aditya}@cs.umd.edu

1 Motivation

On the Semantic Web, the ability to combine, integrate and reuse ontologies is crucial. The Web Ontology Language (OWL) defines the *owl:imports* construct, which allows to include by reference all the axioms contained in another knowledge base (KB) on the Web. This certainly provides some *syntactic* modularity, but not a *logical* modularity. We have proposed [3] $\mathcal{E}$-Connections as a suitable formalism for combining KBs and for achieving modular ontology development on the Web. $\mathcal{E}$-Connections are KR languages defined as a combination of other logical formalisms. They were originally introduced in [4] mostly as a way to go beyond the expressivity of each of the component logics, while preserving the decidability of the reasoning services in the combination. We have found that $\mathcal{E}$-Connections can help process, evolve, reuse, and understand OWL ontologies.

In this paper, we address the problem of automatically transforming an OWL KB $\mathcal{O}$ into a $\mathcal{E}$-Connection Σ in such a way that each of the relevant sub-domains modeled in $\mathcal{O}$ is represented in a different component of Σ. We present a formal definition and investigation of different variants of the problem, a polynomial solution for some of them, an optimized implementation and some promising empirical results.

We have found that in some large KBs, partitioning to an $\mathcal{E}$-Connection provides modularity benefits. In particular, if a KB can be partitioned, it typically contains several "free standing" components, that is, sub-KBs which do not "use" information from any other components. These KBs can be easily reused and evolved without reference to the rest of the $\mathcal{E}$-Connection. We believe that the factoring out of such independent parts of the original KB alone justifies partitioning for many applications.

2 The Partitioning Problem

We first introduce $\mathcal{E}$-Connections as a language for combining $\mathcal{SHOIN}$ KBs.[1]

Definition 1 *Let* $(V_{C_i})_{1\leq i\leq n}$, $(V_{I_i})_{1\leq i\leq n}$, $(\mathcal{E}_{ij})_{1\leq i,j\leq n}$ *be countable and pair-wise disjoint sets of atomic concepts, individuals and relation names respectively. The set of ij-relations, for* $i,j \in \{1,...,n\}$, *is* $\mathcal{E}_{ij} \cup \{P^-|P \in \mathcal{E}_{ji}\}$. *When* $i = j$, *ij-relations are called "roles", whereas if* $j \neq i$ *they are called "link relations". The sets of i-concepts, for* $i \in \{1,...,n\}$ *are built by simultaneous induction as follows:*

$$C := A|\top_i|\neg D|D \sqcap E|D \sqcup E|\{a\}|\exists P.Z|\forall P.Z| \geq nS| \leq nS$$

Where $A \in V_{C_i}$, D, E *are i-concepts,* $a \in V_{I_i}$, Z *is a j-concept and* P, S *ij-relations with* S *simple* [2]. *An i-axiom* $\mathcal{A}$ *is an expression of either of the following forms:*

$$\mathcal{A} := C \sqsubseteq D|P \sqsubseteq Q|Tran(R)|C(a)|P(a,b)$$

With P, Q *ij-relations,* R *an ij-relation with* $i = j$, C, D *i-concepts,* $a \in V_{I_i}, b \in V_{I_j}$.

An $\mathcal{E}$*-Connection* Σ *with vocabulary* $V_n = ((V_{C_i})_{1\leq i\leq n}, (V_{I_i})_{1\leq i\leq n}, (\mathcal{E}_{ij})_{1\leq i,j\leq n})$ *is a collection* $\Sigma = (\Sigma_1, ..., \Sigma_n)$, *where* Σ_i *is a finite set of i-axioms.*

An interpretation is a tuple of the form:

$$\mathcal{M} = \langle (W_i)_{1\leq i\leq n}, (\mathcal{M}_i)_{1\leq i\leq n}, (\mathcal{M}_{ij})_{1\leq i,j\leq n}\rangle$$

Where $W_i \cap W_j = \emptyset, \forall i \neq j$. *The interpretation functions are applied to ij-relations as follows, with* $P \in \mathcal{E}_{ij}, Q \in \mathcal{E}_{ji}$:

$$P^{\mathcal{M}_{ij}} \subseteq W_i \times W_j \mid (Q^-)^{\mathcal{M}_{ij}} = \{(x,y)|(y,x) \in Q^{\mathcal{M}_{ji}}\}$$

For individuals, $a^{\mathcal{M}_i} \in W_i$ *with* $a \in V_{I_i}$. *The interpretation functions are applied to i-concepts and i-axioms as shown in Table 1.* $\mathcal{M}$ *is a model of* Σ ($\mathcal{M} \models \Sigma$) *if* $\forall i = 1,...,n$, $\mathcal{M} \models \mathcal{A}$ *for each i-axiom* $\mathcal{A} \in \Sigma_i$.

The language $\mathcal{C}^{\mathcal{E}}_{\mathcal{IHN}}(\mathcal{SHOIN})$ allows for combinations of $\mathcal{SHOIN}$ ontologies in which inverses, number restrictions and hierarchies are allowed on link relations. This is a language for which we do not have a decision procedure. However, our goal here has been to provide maximum flexibility for the partitioning. Nontheless, there already exist practical (and implemented) tableau-based algorithms for some expressive fragments of this $\mathcal{E}$-Connection language [1], namely $\mathcal{C}^{\mathcal{E}}_{\mathcal{HN}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ and $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$.

[1]For brevity, we give here a slightly simplified version of OWL-DL that does not capture datatypes. However, all the results presented here also apply for combinations of $\mathcal{SHOIN}(\mathcal{D})$ ontologies.

[2]S is *simple* if it is not transitive and none of its sub-relations is transitive

Semantics of i-concepts	Semantics of i-axioms
$A^{\mathcal{M}_i} \subseteq W_i$; $\top_i^{\mathcal{M}_i} = W_i$ $\{a\}^{\mathcal{M}_i} = \{a^{\mathcal{M}_i}\}$	$\mathcal{M} \models (C \sqsubseteq D) \Leftrightarrow C^{\mathcal{M}_i} \subseteq D^{\mathcal{M}_j}$
$(\neg C)^{\mathcal{M}_i} = W_i - C^{\mathcal{M}_i}$	$\mathcal{M} \models (P \sqsubseteq Q) \Leftrightarrow P^{\mathcal{M}_{ij}} \subseteq Q^{\mathcal{M}_{ij}}$
$(C \sqcap D)^{\mathcal{M}_i} = C^{\mathcal{M}_i} \cap D^{\mathcal{M}_i}$	$\mathcal{M} \models Tran(R) \Leftrightarrow (x,y),(y,z) \in R^{\mathcal{M}_{ij}} \Rightarrow (x,z) \in R^{\mathcal{M}_{ij}}$
$(C \sqcup D)^{\mathcal{M}_i} = C^{\mathcal{M}_i} \cup D^{\mathcal{M}_i}$	$\mathcal{M} \models C(a) \Leftrightarrow a^{\mathcal{M}_i} \in C^{\mathcal{M}_i}$
$(\exists P.Z)^{\mathcal{M}_i} = \{x \in W_i \mid \exists y \in Z^{\mathcal{M}_j} \text{ with } (x,y) \in P^{\mathcal{M}_{ij}}\}$	$\mathcal{M} \models P(a,b) \Leftrightarrow (a^{\mathcal{M}_i}, b^{\mathcal{M}_j}) \in P^{\mathcal{M}_{ij}}$
$(\forall P.Z)^{\mathcal{M}_i} = \{x \in W_i \mid \forall y \in W_j, \text{if } (x,y) \in P^{\mathcal{M}_{ij}} \Rightarrow y \in Z^{\mathcal{M}_j}\}$	
$(\leq nS)^{\mathcal{M}_{ij}} = \{x \in W_i \mid \#\{t \in W_j \mid (x,t) \in P^{\mathcal{M}_{ij}}\} \leq n\}$	
$(\geq nS)^{\mathcal{M}_{ij}} = \{x \in W_i \mid \#\{t \in W_j \mid (x,t) \in P^{\mathcal{M}_{ij}}\} \geq n\}$	

Table 1: Semantics of i-Concepts and i-Axioms

Definition 2 *Let $\mathcal{O}$ be a $\mathcal{SHOIN}$ KB with vocabulary $V = (V_C, V_R, V_I)$. The collection $V_n = ((V_{C_i})_{1\leq i\leq n}, (V_{I_i})_{1\leq i\leq n}, (\mathcal{E}_{ij})_{1\leq i,j\leq n})$. is a* ***partitioned vocabulary*** *of V iff:*

- $V_C = \bigcup_{i=1,\ldots,n} V_{C_i}$; $V_I = \bigcup_{i=1,\ldots,n} V_{I_i}$; $V_R = \bigcup_{i,j=1,\ldots,n} \mathcal{E}_{ij}$
- $V_{C_i} \cap V_{C_j} = \emptyset$, $V_{I_i} \cap V_{I_j} = \emptyset$, *for* $i \neq j$ *and* $\mathcal{E}_{ij} \cap \mathcal{E}_{kl} = \emptyset$ *if either* $i \neq k$ *or* $j \neq l$

The first plausible relationship that one could think of between an OWL KB $\mathcal{O}$ and an $\mathcal{E}$-Connection Σ is a "structural" one, i.e. one in which Σ contains exactly the same entities (concepts, properties and individuals) and axioms as $\mathcal{O}$, but divided into its different components.

Definition 3 *Let $\mathcal{O}$ and Σ have respective vocabularies V, V_n. Σ is* ***structurally compatible with*** $\mathcal{O}$ *($\Sigma \sim \mathcal{O}$) iff:*

1. *V_n is a partitioned vocabulary of $\mathcal{O}$*
2. $\mathcal{A} \in \Sigma \Leftrightarrow \mathcal{A} \in \mathcal{O}$

Structurally compatible $\mathcal{E}$-Connections reveal as a plausible output for a partitioning process: it ensures the preservation in the $\mathcal{E}$-Connection of the modeling choices in the original ontology, since no entities or axioms are added, removed or changed. However, structural compatibility does not provide semantic guarantees, since many relevant entailments that held in the original ontology might not hold anymore in the output $\mathcal{E}$-Connection. The reason is that the semantics of $\Sigma \sim \mathcal{O}$ differs from the semantics of $\mathcal{O}$. For example, suppose that the negation $(\neg C)$ is present in $\mathcal{O}$ and also in Σ_i, for $\Sigma = (\Sigma_1, ..., \Sigma_n)$ with $\Sigma \sim \mathcal{O}$. If $\mathcal{I} = (W, .^{\mathcal{I}})$ is an interpretation of $\mathcal{O}$, and $\mathcal{M} = \langle (W_i)_{1\leq i\leq n}, (\mathcal{M}_i)_{1\leq i\leq n}, (\mathcal{M}_{ij})_{1\leq i,j\leq n}\rangle$ an interpretation of Σ, then:

$$(\neg C)^{\mathcal{I}} = W - C^{\mathcal{I}} \quad ; \quad (\neg C)^{\mathcal{M}} = W_i - C^{\mathcal{M}_i}$$

Observe that $\mathcal{I}$ computes the set difference w.r.t. the whole interpretation domain, whereas $\mathcal{M}$ does it w.r.t. the restricted "local" domain W_i. Similar variations in the semantics are revealed in other constructs.

In order to determine how these differences affect the relationship between $\mathcal{O}$ and Σ, we will compare "corresponding" interpretations, i.e. interpretations that "agree" on atomic concepts, properties and individuals, but "disagree" in general in the evaluation of complex constructs.

Definition 4 *Let $\mathcal{O}$ be an KB with vocabulary V and let V_n be a partitioned vocabulary for V. A* ***partitioned interpretation*** *of $\mathcal{O}$ with vocabulary V_n, denoted by $I(V_n) = (W, .^{\mathcal{I}(V_n)})$ is an interpretation for $\mathcal{O}$ of the form:*

- $W = \bigcup_{i=1,\ldots,n} W_i$ *with* $W_i \cap W_j = \emptyset$ *for* $i \neq j$, *and* $W_i \neq \emptyset$
- $A^{\mathcal{I}(V_n)} \subseteq W_i$, *for each* $A \in V_{C_i}$
- $P^{\mathcal{I}(V_n)} \subseteq W_i \times W_j$, *for each* $P \in \mathcal{E}_{ij}$
- $a^{\mathcal{I}(V_n)} \in W_i$, *for each* $a \in V_{I_i}$

We say that $\mathcal{O}$ is ***partitionable for*** *V_n if there exists an interpretation $I(V_n)$ s.t. $I(V_n) \models \mathcal{O}$.*

Definition 5 *Let $\mathcal{O}$ and Σ have vocabularies V and V_n respectively, with V_n a partitioned vocabulary for V. We can establish a relation '$\leftrightarrow$' between the interpretations of Σ and the partitioned interpretations of $\mathcal{O}$ with vocabulary V_n s.t. $I(V_n) \leftrightarrow \mathcal{M}$ are related as follows:*

- $W_i' = W_i$; $\top^{\mathcal{I}(V_n)} = \bigcup_i (\top_i)^{\mathcal{M}}$
- $A^{\mathcal{I}(V_n)} = A^{\mathcal{M}_i}$, *for each* $A \in V_{C_i}$
- $P^{\mathcal{I}(V_n)} = P^{\mathcal{M}_{ij}}$, *for each* $P \in \mathcal{E}_{ij}$
- $a^{\mathcal{I}(V_n)} = a^{\mathcal{M}_i}$, *for each* $a \in V_{I_i}$

Note that the relation '$\leftrightarrow$' is a bijection. We are now ready to formulate the notion of semantic compatibility between a DL KB and an $\mathcal{E}$-Connection:

Definition 6 *Let $\mathcal{O}$ and Σ have respective vocabularies V and V_n, with V_n a partitioned vocabulary for V, then Σ is* ***semantically compatible*** *with $\mathcal{O}$ ($\Sigma \approx \mathcal{O}$) if:*

1. *$\mathcal{O}$ is partitionable for V_n*
2. *If $\mathcal{I}(V_n) \leftrightarrow \mathcal{M}$, then $\mathcal{M} \models \Sigma$ iff $\mathcal{I}(V_n) \models \mathcal{O}$*

Semantic compatibility is a desirable relation between the input and the output of a partitioning process. It ensures that equivalent KBs have exactly the same set of compatible $\mathcal{E}$-Connections. It preserves consistency and ensures that existing subsumptions in the class tree and property tree will hold in the $\mathcal{E}$-Connection (see [2] for details). As an example of structural and semantic compatibility, suppose the following KB:

$$\mathcal{O} = \{(C \sqsubseteq D \sqcup E); (C \sqsubseteq \neg D); (B \sqsubseteq \exists P.A); (A \sqsubseteq F)\}$$

and the $\mathcal{E}$-Connections $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$ and $\Upsilon = (\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4)$:

$$\Sigma_1 = \{C \sqsubseteq D \sqcup E, C \sqsubseteq \neg D\}; \Sigma_2 = \{B \sqsubseteq \exists P.A\}; \Sigma_3 = \{A \sqsubseteq F\}$$
$$\Upsilon_1 = \{C \sqsubseteq E\}\ ; \Upsilon_2 = \{B \sqsubseteq \exists P.A\}; \Upsilon_3 = \{A \sqsubseteq F\}; \Upsilon_4 = \{D \sqsubseteq \bot\}$$

Observe that $\Sigma \sim \mathcal{O}$ and $\Upsilon \approx \mathcal{O}$. We can formulate at least three partitioning problems depending on what is the desired relationship between the input $\mathcal{O}$ and the output Σ.

Definition 7 *The partitioning problem* ***P1)*** *is the problem of finding, for an ontology $\mathcal{O}$, the $\mathcal{E}$-Connection Σ with the largest number of components s.t. $\Sigma \sim \mathcal{O}$. In the problem* ***P2)*** *we require $\Sigma \approx \mathcal{O}$* ***in addition to*** *$\Sigma \sim \mathcal{O}$. In* ***P3)*** *we require $\Sigma \approx \mathcal{O}$* ***instead of*** *$\Sigma \sim \mathcal{O}$.*

In other words, we can enforce structural compatibility only, both structural and semantic compatibility, or semantic compatibility only.

In this paper, we show that **P1)** and **P2)** are solvable in polynomial time, without the intervention of a reasoner in any stage of the process. We leave the solution of **P3)** as an open problem. The key step toward a solution for **P2)** is to devise under what conditions $\Sigma \sim \mathcal{O}$ is also semantically compatible with $\mathcal{O}$. For such analysis, the notion of **$\mathcal{E}$-safety** reveals crucial.

Definition 8 *Let: $g : C \in \mathcal{O} \rightarrow \{T, F\}$ be a function mapping every concept $C \in \mathcal{O}$ to a boolean value and recursively defined as follows:*

- *If C is $\top$, then $g(C) = F$*
- *Let $C \in V_C$, then $g(C) = T$*
- *Let C be $\{a\}$, for $a \in V_I$, then $g(C) = T$*
- *Let C be $D \sqcap E$.If $g(D)$=F and $g(E)$=F, then $g(C)$=F. Otherwise, $g(C) = T$*
- *Let C be $D \sqcup E$. If $g(D) = T$ and $g(E) = T$, then $g(C) = T$. Otherwise, $g(C) = F$*
- *Let C be $\neg D$. If $g(D) = T$, then $g(C) = F$ and if $g(D) = F$, then $g(C) = T$.*
- *Let C be $\exists P.D$ or $\geq nP$, then $g(C) = T$*
- *Let C be $\forall P.D$ or $\leq nP$, then $g(C) = F$*

The KB $\mathcal{O}$ is ***$\mathcal{E}$-safe*** *iff it contains no axiom of the form $C \sqsubseteq D$ s.t. $g(C) = F$ and $g(D) = T$.*

The function g determines which concepts are interpreted in the same way by corresponding interpretations ($g(C) = T$) or differently ($g(C) = F$), due to the differences between DL and $\mathcal{E}$-Connections semantics. $\mathcal{E}$-safety is a property of the input ontology that indicates which axioms are "dangerous" for the preservation of semantic compatibility in an $\mathcal{E}$-Connections based decomposition (see [2] for details).

The exact connection between structural and semantical compatibility is given by the following theorem (see [2] for a proof):

Theorem 1 *If $\mathcal{O}$ is consistent and $\mathcal{E}$-safe and $\Sigma \sim \mathcal{O}$, then $\Sigma \approx \mathcal{O}$. If $\mathcal{O}$ is not $\mathcal{E}$-safe, then the only Σ s.t. $\Sigma \sim \mathcal{O}$ and $\Sigma \approx \mathcal{O}$ is $\Sigma = (\mathcal{O})$.*

The theorem shows how **P2)** can be reduced to **P1)**. In order to solve **P2)**, first decide $\mathcal{E}$-safety and then solve **P1)**.Obviously, $\mathcal{E}$-safety can be computed efficiently and hence the reduction is polynomial.

3 The Partitioning Algorithm

In this section, we briefly describe our proposed algorithm for solving **P1)** and **P2)**. For a detailed description and specification we refer to [2]. The algorithm accepts $\mathcal{O}$ as input and returns an $\mathcal{E}$-Connection $\Sigma = (\Sigma_1, ..., \Sigma_n)$. The algorithm consists of a succession of *n partitioning steps*. Each step involves *a pair* of KBs: the *original* KB, $\mathcal{O}$, from which entities and axioms are removed, and a *target* KB, Σ_i, generated from scratch in the ith step, to which these are added. In the process some roles will eventually become link relations.

The algorithm initially checks if $\mathcal{O}$ is $\mathcal{E}$-safe. If the check is negative, then it returns $\mathcal{O}$ as a result. Otherwise, the algorithm starts a partitioning step by creating a new component Σ_i and by forcing an initial state transition on an arbitrary entity (concept, role or individual) in $\mathcal{O}$.

The initial transition will trigger new ones, due to the structural constraints imposed by $\mathcal{E}$-Connections. For example, if $(C \sqsubseteq D) \in \mathcal{O}$, and we move C, then D *must* be exported as well to Σ_i, since an axiom cannot relate complex classes in different components in an $\mathcal{E}$-Connection. However, there is a choice in certain situations that involve roles. For example, if $\exists R.C \in \mathcal{O}$ and we move C, two possible actions would be allowed: first, to make R a link relation from $\mathcal{O}$ to Σ_i; second, to make R a role in Σ_i. Analogously, if $P(a, b) \in \mathcal{O}$, and we move a we can transform P into link relation from Σ_i to $\mathcal{O}$, or into role in Σ_i. In both examples, each choice would result in a syntactically valid $\mathcal{E}$-Connection. In order to obtain a maximal partitioning of $\mathcal{O}$ we will transform roles into link relations *whenever possible*.

In a given partitioning step, each concept, individual and link relation (generated in a previous step) can be in one of two possible "states": either as entities in $\mathcal{O}$, or in Σ_i. A role, however, can be in one of four possible states: as a role in $\mathcal{O}$(State 1), as a link relation from $\mathcal{O}$ to Σ_i (State 2), as a link relation from Σ_i to $\mathcal{O}$ (State 3), and finally as a role in Σ_i (State 4). Only the transitions $1 \rightarrow 2$, $1 \rightarrow 3$, $1 \rightarrow 4$ $2 \rightarrow 4$ and $3 \rightarrow 4$ are allowed.

Once all the state transitions in a partitioning step have been completed, the relevant axioms are moved. Each axiom in the input ontology is moved *only once*, i.e. whenever it is exported from $\mathcal{O}$ to a newly created component, it will never be put back into $\mathcal{O}$, nor moved to a different component.

Theorem 2 *The algorithm Partition(O) is worst-case quadratic in the size of the KB. The output Σ is a solution for **P2)** with input $\mathcal{O}$ and a solution for **P1)** if the safety check is omitted.*

KB	Atomic Concepts	Complex Descriptions	Roles	Indiv.	Number Components/ Leaf Comp.	Atomic Concepts Largest	Atomic Concepts Smallest	Links in Σ	Time(s)
OWL-S	51	49	54	9	17/7	21	1	37	0.291
NASA	1537	232	102	194	43/36	1100	1	22	2.8
GALEN	2749	2011	413	0	2/1	2748	1	0	11
NCI	27652	4000	71	0	17/9	7663	34	55	45

Table 2: Some Partitioned Ontologies

4 Implementation and Evaluation

We have implemented our partitioning algorithm on top of Manchester's OWL-API, which we have extended to provide support for $\mathcal{E}$-Connections. The UI in SWOOP [3] for browsing $\mathcal{E}$-Connections has been extended to support automated partitioning. We have applied our algorithm to a set of OWL ontologies available on the Web and stored the partitions in an online repository [4]. Table 2 summarizes the results obtained for some relevant cases.

GALEN and NCI are both large, carefully designed ontologies dealing with the biomedical domain, but which follow very different modeling paradigms. In GALEN, most of the knowledge is ultimately depending on a common *top* class and *top* role. Hence, although it is possible to identify intuitively several disjoint sub-domains in GALEN, the ontology follows a very "monolithic" design pattern, which prevents a good partitioning. However, NCI follows a more modular design pattern, since the knowledge has been structured around separate top entities, each of which defines a different sub-domain.

The partitioning of NCI reproduces each of the intuitive sub-domains in a different component. The link relations in the resulting $\mathcal{E}$-Connection provide useful information on the original ontology. In the partitioning of NCI the component dealing with genes is the one that contains the largest number of "outgoing" link relations and also the one that "uses" information from the largest number of components. This implies that genes are central to the ontology. Other components, like the one dealing with anatomical structures, are "leaf components" in the sense that they have only "incoming" link relations. These components do not use information from any other components in the $\mathcal{E}$-Connection, are written in plain OWL and can be directly reused.

The OWL-S ontologies describe Web services, whereas NASA's SWEET-JPL ontologies model several interrelated domains of interest for the space industry. Within both sets of KBs, each ontology seems to model a well-defined sub-domain. The domain as a whole is modeled in both cases by using *owl:imports*. We have collapsed each set of ontologies applied the partitioning algorithm to the result. In the case of OWL-S the partitions correspond closely to the original

[3]http://www.mindswap.org/2004/SWOOP

[4]http://www.mindswap.org/2004/multipleOnt/FactoredOntologies

sub-domains: no information from different KBs in the original set comes together, while some of the original sub-domains appear further decomposed in a reasonable way. However, in the case of NASA-JPL, the partitioning shows important flaws in the way the knowledge was originally structured. The physically distinct ontologies do not correspond to semantically distinct ones.

5 Related Work and Conclusion

In this paper, we have analyzed the problem of partitioning expressive DL KBs using $\mathcal{E}$-Connections. We have argued that partitioning can be a useful tool for Semantic Web applications. We have provided an efficient solution for some interesting partitioning problems and shown some empirical results.

Partitioning OWL ontologies for modeling purposes has been recently addressed in [6]. We consider that the main limitation of that approach is the lack of a suitable formalism for representing the output of the problem. The results are represented as a visualization of the different kinds of information contained in the input ontology. No shareable partitions are obtained.

[5] explores partitioning FOL theories to improve theorem prover performance. We believe that reasoning $\mathcal{E}$-Connections not only does not affect existing optimizations in DL reasoners, but also suggest new ones. In particular it may help to detect obvious non-subsumptions, to alleviate the effect of non-absorbable GCIs and to enhance ABox reasoning (see [3] for details). We plan to confirm that experimentally using our reasoner Pellet, which already provides $\mathcal{E}$-Connections support and compare the results with [5] for the DL case.

References

[1] B. Cuenca-Grau, B. Parsia, and E.Sirin. Tableau algorithms for $\mathcal{E}$-Connections of description logics. Technical report, UMIACS, 2004. http://www.mindswap.org/2004/multipleOnt/papers/EconnTableau.ps.

[2] B. Cuenca-Grau, B. Parsia, E.Sirin, and A.Kalyanpur. Automatic partitioning of OWL ontologies using $\mathcal{E}$-Connections. Technical report, UMIACS, 2005. Available at http://www.mindswap.org/2004/multipleOnt/papers/Partition.pdf.

[3] B. Cuenca Grau, B. Parsia, and E.Sirin. Combining owl ontologies using $\mathcal{E}$-connections. *Elsevier's Journal on Web Semantics*, 2005. To Appear. http://www.mindswap.org/2004/multipleOnt/papers/EconnJWS.pdf.

[4] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-Connections of abstract description systems. *Artificial Intelligence 156(1):1-73*, 2004.

[5] B. MacCartney, S. A. McIlraith, E. Amir, and T. Uribe. Practical partition-based theorem proving for large knowledge bases. In *Proc. of IJCAI 2003*, 2003.

[6] H. StuckenSchmidt and M. Klein. Structure-based partitioning of large class hierarchies. In *Proc. of ISWC 2004*, 2004.