UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

DISTRIBUTED REASONING SERVICES FOR MULTIPLE ONTOLOGIES

Luciano Serafini and Andrei Tamilin

May 2004

Technical Report # DIT-04-029

Distributed Reasoning Services for Multiple Ontologies *

Luciano Serafini[1] and Andrei Tamilin[2]

[1] ITC-IRST, Trento 38050, Italy,
luciano.serafini@itc.it,
[2] DIT, University of Trento, Trento 38050, Italy,
andrei.tamilin@dit.unitn.it

Abstract. The main goal of this paper is to propose a distributed paradigm for reasoning with multiple ontologies connected by semantic mappings. The contribution of the paper to this goal is twofold. From the theoretical point of view we characterize the problem of global subsumption (i.e. the problem of subsumption in a set of local ontologies connected by semantic mappings) as a suitable fixpoint combination of operators that compute subsumptions in the local ontologies. This allows us to define a sound and complete algorithm for global subsumptions which calls black-boxes sub-routines for local subsumptions. The second contribution is the description of a prototype implementation of such algorithm in a peer-to-peer architecture.

1 Introduction

Ontologies have been advocated as a means of interoperability support between distributed applications and web services. The basic idea is that different autonomously developed applications can meaningfully communicate by using a common repository of meaning, i.e. a shared ontology. The optimal solution obviously lies in having a unique worldwide shared ontology describing all possible domains. Unfortunately, this is unachievable in practice. The actual situation is characterized by a proliferation of different ontologies. Existing ontologies vary in a number of respects, such as the level of granularity of the semantic information they provide, the extension of their conceptual coverage, the size of the domain they describe, and the perspective from which such a domain is described. The level of granularity, the domain and the perspective of an ontology are parameters that localize the ontology in an imaginary ontology space. To stress this fact we will use the term *local ontologies*. Furthermore, in the current situation, the same domain can be described by different ontologies in a heterogeneous way. So, the very same concept can be described in a different way and at a different level of detail by more than one ontology. To stress this fact we will use the term *overlapping ontologies*. Semantic interoperability, therefore, can be solved by discovering the *semantic mappings* between the concepts

* We thank Floris Roelofsen for proof readings and comments and Mark Carman for proof readings.

defined in the local ontologies. Many efforts have been performed to develop techniques for finding semantic mappings (See "The Ontology Alignment Source " at `http://www.atl.external.lmco.com/projects/ontology` for more information on this topic).

A set of local ontologies related via semantic mappings are not enough to guarantee interoperability. One has to provide also the capability of reasoning in such a system. *Reasoning services* are necessary for *checking consistency* of semantic mappings, or for *discovering new mappings* such as, for instance, the composition of existing mappings. Furthermore, semantic mappings allow one to *transfer ontological knowledge* between ontologies. Reasoning services should compute new ontological properties that derive from the combination of ontologies with semantic mappings.

The state of the art methodology for reasoning with multiple ontologies is based on the idea of combining all ontologies and mappings into a common *global ontology* [2, 20], and reasoning in this global ontology using powerful description logic reasoners, such as RACER [8] and FaCT [9].

This approach, however, can not scale to the whole web because of three main problems. The first one is a computational problem. Clearly, the search space used for reasoning in the global ontology is much larger than the sum of the search spaces used for reasoning in local ontologies. Thus global reasoning is likely to be less efficient than a suitable combination of local reasonings. The second problem concerns the specificity of reasoning. Reasoning in local ontologies can be done by specific reasoners which are optimized for the local language. Reasoning in the global ontology has to be performed by the most general reasoner which is capable of dealing with the most general local language. The third problem comes from aspects of information hiding. In some cases, local ontologies as a whole are not available. The access to such ontologies is limited to a query interface that, for instance, allows one only to submit queries of limited form.

The main goal of this paper is to propose an alternative approach, which is a *distributed* paradigm for reasoning with multiple ontologies connected by semantic mappings. The contribution of the paper to this goal is twofold. From the theoretical point of view we characterize the problem of *global subsumption* (i.e. the problem of subsumption in a set of local ontologies connected by semantic mappings) as a suitable fixpoint combination of operators that compute subsumptions in the local ontologies. This allows us to define a sound and complete algorithm for computing global subsumptions which calls black-boxes subroutines for computing local subsumptions. The second contribution is the description of a prototype implementation of the algorithm in a peer-to-peer architecture.

The theoretical development is based on the long tradition of logics for distributed systems, based on Multi-Context Systems [6, 7] and its Local Models Semantics [4], the extension of First Order Logics which leads to Distributed First Order Logic [5], and extensions to Description Logics which leads to Distributed Description Logics (DDL) [2]

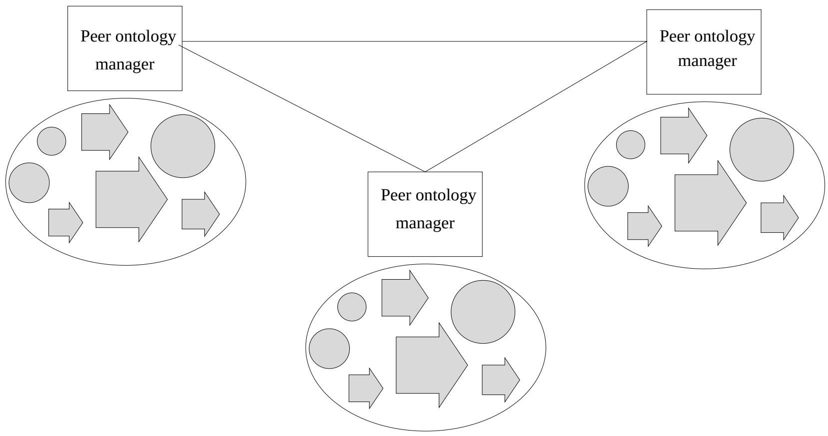

Fig. 1. P2P architecture for managing multiple ontologies. In each peer, circles represent ontologies and arrows represent semantic relations (i.e. mappings) between ontologies.

From the architectural point of view, we have been inspired by peer-to-peer (P2P) distributed knowledge management architectures, proposed in the Edamok [14] projects, and by the C-OWL language [3]. In particular, we have implemented a peer-to-peer architecture, whose schema is shown in Figure 1, consisting of *peer ontology managers* capable of providing reasoning services on a set of local ontologies, and capable of requesting reasoning services to the other ontology managers. The ontology manager of a peer p is capable of providing *local* and *global* ontology services. Local services involve only ontologies that are local to p, while global services involve both ontologies in p and in other semantically related peers. Among the provided reasoning services, the most important and fundamental ones are checking a local and a global subsumption.

The rest of the paper is organized as follows. In section 2 we describe the DDL framework as introduced in [2]; in section 3 we state our results of the DDL analysis, introduce a general intuition for computing global subsumption, and describe a sound and complete distributed reasoning algorithm built as a composition of existing local reasoning algorithms; in section 4 we overview briefly our java-based prototype implementation of a peer ontology manager that incapsulates the developed algorithm, and is integrated into the Protégé-2000 ontology editing tool [16]; section 5 gives the related work, and finally, we conclude in section 6.

2 Distributed Description Logics

The main purpose of *Distributed Description Logics (DDL)*, defined by Borgida and Serafini in [2], is to provide a syntax and semantics, formalizing the case of multiple overlapping local ontologies pairwise linked by semantic mappings.

In DDL, local ontologies are represented by description logic theories (T-boxes), while semantic mappings are represented by *bridge rules*. In this section we briefly recall the definition of DDL as given in [2].

2.1 Syntax

Given a non empty set I of indexes, used to enumerate local ontologies, let $\{\mathcal{D}L_i\}_{i \in I}$ be a collection of description logics. Each $\mathcal{D}L_i$ can be one of the logics which is weaker or equivalent to $\mathcal{SHIQ}$[11] (e.g., $\mathcal{ALC}$, $\mathcal{ALCN}$, $\mathcal{SH}$)[3]. For each $i \in I$ let us denote a T-box of $\mathcal{D}L_i$ as $\mathcal{T}_i$. We call $\mathbf{T} = \{\mathcal{T}_i\}_{i \in I}$ a family of T-Boxes on I. Intuitively, $\mathcal{T}_i$ is the description logic formalization of the i-th ontology. To make every description distinct, we will prefix it with the index of ontology it belongs to. For instance, the concept C that occurs in the i-th ontology is denoted as $i : C$. Similarly, $i : C \sqsubseteq D$ denotes the fact that the axiom $C \sqsubseteq D$ is stated in the i-th ontology.

Semantic mappings between different ontologies are expressed via *bridge rules*.

Definition 1 (Bridge rules). *A bridge rule from i to j is an expression of the following two forms:*

1. $i : x \xrightarrow{\sqsubseteq} j : y$, *an* into-bridge rule
2. $i : x \xrightarrow{\sqsupseteq} j : y$, *an* onto-bridge rule

where x and y are either two concepts, or two roles, or two individuals of $\mathcal{D}L_i$ and $\mathcal{D}L_j$ respectively.

Bridge rules do not represent semantic relations stated from an external *objective* point of view. Indeed, there is no such global view in the web. Instead, bridge rules from i to j express relations between i and j as viewed from the *subjective* point of view of the j-th ontology. Intuitively, the into-bridge rule $i : C \xrightarrow{\sqsubseteq} j : D$ states that, from the j-th point of view the concept C in i is less general than its local concept D. Similarly, the onto-bridge rule $i : C \xrightarrow{\sqsupseteq} j : D$ expresses the fact that, according to j, C in i is more general than D in j. Therefore, bridge rules from i to j represent the possibility of j's ontology to translate (under some approximation) the concepts of foreign i's ontology into its internal model. Note, that since bridge rules reflect the subjective point of view, bridge rules from j to i are not necessarily the inverse of the rules from i to j.

Definition 2 (Distributed T-box). *A distributed T-box (DTB)* $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ *consists of a collection of T-boxes* $\{\mathcal{T}_i\}_{i \in I}$*, and a collection of bridge rules* $\mathfrak{B} = \{\mathfrak{B}_{ij}\}_{i \neq j \in I}$ *between them.*

[3] We assume familiarity with description logics and related tableaux-based reasoning systems described in [11].

Example 1. The International Standard Classification of Occupations (ISCO-88)[4] is an ontology that organizes occupations in a hierarchical framework. At the lowest level there is the unit of classification–a job–which is defined as a set of tasks or duties designed to be executed by one person. Jobs are grouped into occupations according to the degree of similarity in their constituent tasks and duties. Occupations are grouped by skill levels. An extract from ISCO-88 is shown on the left side of Figure 2.

```
                          ISCO-88                                                 WordNet
2 Professionals                                                            adEntity
 21 Physical, mathematical and engineering science professionals            Causal_agency
  211 Physicists, chemists and related professionals                         Cause
   2111 Physicists and astronomers                                            Causal_agent
   2114 Geologists and geophysicists                                           Entity
  212 Mathematicians, statisticians and related professionals                   Physical_object
   2121 Mathematicians and related professionals                                 Object
   2122 Statisticians                                                             Animate_thing
  213 Computing professionals                                                      Living_thing
   2131 Computer systems designers, analysts and programmers                        Being
   2139 Computing professionals not elsewhere classified                             Organism
  214 Architects, engineers and related professionals                                 Person
   2141 Architects, town and traffic planners                                          Self
   2146 Chemical engineers                                                              Grownup
3 Technicians and associate professionals                                               Adventurer
 31 Physical and engineering science associate professionals                            Nurser
  311 Physical and engineering science technicians                                      Engineer
   3111 Chemical and physical science technicians                                       Capitalist
   3112 Civil engineering technicians                                                   Captor
   3119 Physical and eng. science technicians not elsewhere classif.                    Commoner
  312 Computer associate professionals                                                  Worker
```

Fig. 2. An extract from ISCO-88 and WordNet.

A similar, though less detailed, ontology of occupations can be found in the "People" subhierarchy of WordNet[5] (is shown on the right side of Figure 2). WordNet provides a plain list of terms without distinguishing between broader terms such as "worker" and narrower terms such as "engineer".

If one wants to enrich WordNet with the classification described in ISCO without modifying WordNet itself, a set of bridge rules from ISCO to WordNet would provide a possible solution. Bridge rules allow to state semantic correspondences between the concepts defined in two ontologies, inducing new subsumptions between WordNet concepts. Examples of bridge rules are the following:

$$\mathrm{ISCO} : \mathsf{Professionals} \xrightarrow{\sqsubseteq} \mathrm{WNP} : \mathsf{Worker} \tag{1}$$

$$\mathrm{ISCO} : \mathsf{Technicians\_And\_Associate\_Professionals} \xrightarrow{\sqsubseteq} \mathrm{WNP} : \mathsf{Worker} \tag{2}$$

$$\mathrm{ISCO} : \begin{array}{r}\mathsf{Architects\_Engineers\_And\_Related\_Professionals} \sqcup \\ \mathsf{Physical\_And\_Engineering\_Science\_Associate\_Professionals}\end{array} \xrightarrow{\sqsupseteq} \mathrm{WNP} : \mathsf{Engineer} \tag{3}$$

$$\mathrm{ISCO} : \top \xrightarrow{\sqsubseteq} \mathrm{WNP} : \neg\mathsf{Child} \tag{4}$$

[4] http://www.ilo.org/public/english/bureau/stat/class/isco.htm

[5] http://xmlns.com/wordnet/1.6/Person

$$\text{ISCO} : \mathsf{Doorkeepers\_watchpersons\_and\ldots} \xrightarrow{\sqsupseteq} \text{WNP} : \mathsf{Gatekeeper} \quad (5)$$

2.2 Semantics

DDL semantics is a customization of the Local Models Semantics for Multi Context Systems [4, 5]. The basic idea is that each ontology $\mathcal{T}_i$ is *locally interpreted* on a *local domain*. The first component of the semantics of a DTB is therefore a family of interpretations $\{\mathcal{I}_i\}_{i\in I}$, one for each T-box $\mathcal{T}_i$. Each $\mathcal{I}_i$ is called a *local interpretation*.

In order to deal with ontologies which are locally unsatisfiable (this can happen when local axioms are not satisfiable or when bridge rules with other ontologies are not satisfiable) we will introduce two special types of local interpretations called *holes*.

Definition 3 (Holes). *A full hole in a T-box $\mathcal{T}$ is an interpretation $\mathcal{I}^{\Delta} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}^{\Delta}} \rangle$, where $\Delta^{\mathcal{I}}$ is the original nonempty domain in $\mathcal{T}$, and $\cdot^{\mathcal{I}^{\Delta}}$ is a function that maps every concept expression in $\mathcal{T}$ in the whole $\Delta^{\mathcal{I}}$. An empty hole in $\mathcal{T}$ as an interpretation $\mathcal{I}^{\emptyset} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}^{\emptyset}} \rangle$, where $\Delta^{\mathcal{I}}$ is the original nonempty domain $\mathcal{T}$, and $\cdot^{\mathcal{I}^{\emptyset}}$ is a function that maps every concept expression in $\mathcal{T}$ in the empty set.*

According to the above definition, holes interpret every concept, both atomic and complex ones, either in the empty set or in the universe. The recursive definition of concept interpretation is not applied for holes. One should notice that the interpretation of concept $(\neg C)^{\mathcal{I}_{\emptyset}}$ is not $\Delta^{\mathcal{I}_{\emptyset}} \setminus C^{\mathcal{I}_{\emptyset}} = \Delta^{\mathcal{I}_{\emptyset}}$, but it is $\emptyset$. The consequence of this fact is that $\mathcal{I}_{\emptyset} \models C \sqsubseteq D$ and $\mathcal{I}_{\Delta} \models C \sqsubseteq D$ for any pair of concepts C and D. Obviously, since both $\mathcal{I}^{\Delta}$ and $\mathcal{I}^{\emptyset}$ satisfy all (even contradictory) concepts in $\mathcal{T}$, they are models of $\mathcal{T}$, i.e. $\mathcal{I}^{\Delta} \models \mathcal{T}$ and $\mathcal{I}^{\emptyset} \models \mathcal{T}$.

Holes provide semantics to a set of ontologies, where some of them are inconsistent. For instance, the distributed interpretation of the ontologies $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}_3$, where $\mathcal{T}_2$ is inconsistent, is a triple $\langle \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3 \rangle$, where $\mathcal{I}_2$ is a hole.

Ontology interpretations can be defined over heterogeneous domains (e.g., representation of time in two ontologies can be done in different domains: one in the domain of Rationals and another in the domain of Naturals respectively). We therefore need a set of relations that model semantic correspondences between heterogeneous domains.

Definition 4 (Domain relation). *A domain relation r_{ij} from $\Delta^{\mathcal{I}_i}$ to $\Delta^{\mathcal{I}_j}$ is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$, We use $r_{ij}(d)$ to denote $\{d' \in \Delta^{\mathcal{I}_j} \mid \langle d, d' \rangle \in r_{ij}\}$; for any subset D of $\Delta^{\mathcal{I}_i}$, we use $r_{ij}(D)$ to denote $\bigcup_{d\in D} r_{ij}(d)$; for any $R \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ we use $r_{ij}(R)$ to denote $\bigcup_{\langle d,d' \rangle \in R} r_{ij}(d) \times r_{ij}(d')$.*

As in the case of bridge rules, domain relation r_{ij} does not represent a semantic mapping seen from an external objective point of view. Rather, it represents a possible way of mapping the elements of $\Delta^{\mathcal{I}_i}$ into its domain $\Delta^{\mathcal{I}_j}$, seen from j's perspective. For instance, if $\Delta^{\mathcal{I}_1}$ and $\Delta^{\mathcal{I}_2}$ are the representation of time on Rationals and on Naturals, r_{ij} could be the round off function, or an analogous approximation relation.

Definition 5 (Distributed interpretation). *A distributed interpretation* $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I} \rangle$ *of distributed T-boxes* $\mathfrak{T}$ *consists of local interpretations* $\mathcal{I}_i$ *for each* $\mathcal{T}_i$ *on local domains* $\Delta^{\mathcal{I}_i}$*, and a family of domain relations* r_{ij} *between these local domains.*

Definition 6. *A distributed interpretation* $\mathfrak{I}$ *satisfies (written* $\mathfrak{I} \vDash_d$*) the elements of a DTB* $\mathfrak{T}$ *according to the following clauses: for every* $i, j \in I$

1. $\mathfrak{I} \vDash_d i : A \sqsubseteq B$*, if* $\mathcal{I}_i \vDash A \sqsubseteq B$
2. $\mathfrak{I} \vDash_d \mathcal{T}_i$*, if* $\mathfrak{I} \vDash_d i : A \sqsubseteq B$ *for all* $A \sqsubseteq B$ *in* $\mathcal{T}_i$
3. $\mathfrak{I} \vDash_d i : x \xrightarrow{\sqsubseteq} j : y$*, if* $r_{ij}(x^{\mathcal{I}_i}) \subseteq y^{\mathcal{I}_j}$
4. $\mathfrak{I} \vDash_d i : x \xrightarrow{\sqsupseteq} j : y$*, if* $r_{ij}(x^{\mathcal{I}_i}) \supseteq y^{\mathcal{I}_j}$
5. $\mathfrak{I} \vDash_d \mathfrak{B}_{ij}$*, if* $\mathfrak{I}$ *satisfies all bridge rules in* $\mathfrak{B}_{ij}$
6. $\mathfrak{I} \vDash_d \mathfrak{T}$*, if for every* $i, j \in I$*,* $\mathfrak{I} \vDash_d \mathcal{T}_i$ *and* $\mathfrak{I} \vDash_d \mathfrak{B}_{ij}$*.*
7. $\mathfrak{T} \vDash_d i : C \sqsubseteq D$ *if for every* $\mathfrak{I}$*,* $\mathfrak{I} \vDash_d \mathfrak{T}$ *implies* $\mathfrak{I} \vDash_d i : C \sqsubseteq D$*.*

2.3 Properties

In this section we show the basic properties of DDL. Hereafter, $\mathfrak{B}_{ij}^{\text{into}}$ and $\mathfrak{B}_{ij}^{\text{onto}}$ will denote the set of into- and onto-bridge rules of $\mathfrak{B}_{ij}$ respectively.

Monotonicity Bridge rules do not delete local subsumptions. Formally:

$$\mathcal{T}_i \vDash A \sqsubseteq B \Longrightarrow \mathfrak{T} \models_d i : A \sqsubseteq B \tag{6}$$

Directionality T-box without incoming bridge rules is not affected by other T-boxes. Formally, if $\mathfrak{B}_{ki} = \emptyset$ for any $k \neq i \in I$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \Longrightarrow \mathcal{T}_i \vDash A \sqsubseteq B \tag{7}$$

Strong directionality Sole into- or sole onto-bridge rules incoming to local terminology do not affect it. Formally, if for all $k \neq i$ either $\mathfrak{B}_{ki}^{\text{into}} = \emptyset$ or $\mathfrak{B}_{ki}^{\text{onto}} = \emptyset$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \Longrightarrow \mathcal{T}_i \vDash A \sqsubseteq B \tag{8}$$

Local inconsistency The fact that $\mathfrak{B}_{ij}$ contains into- and onto-bridge rules does not imply that inconsistency propagates. Formally:

$$\mathfrak{T} \models_d i : \top \sqsubseteq \bot \not\Longrightarrow \mathfrak{T} \models_d j : \top \sqsubseteq \bot \tag{9}$$

Simple subsumption propagation Combination of onto- and into-bridge rules allows to propagate subsumptions across ontologies. Formally, if $\mathfrak{B}_{ij}$ contains $i : A \xrightarrow{\sqsupseteq} j : G$ and $i : B \xrightarrow{\sqsubseteq} j : H$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \Longrightarrow \mathfrak{T} \models_d j : G \sqsubseteq H \tag{10}$$

Generalized subsumption propagation If $\mathfrak{B}_{ij}$ contains $i : A \xrightarrow{\sqsupseteq} j : G$ and $i : B_k \xrightarrow{\sqsubseteq} j : H_k$ for $1 \leq k \leq n$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq \bigsqcup_{k=1}^{n} B_k \Longrightarrow \mathfrak{T} \models_d j : G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \tag{11}$$

Let us prove properties (9) and (11). The first one is important as it allows us to explain how full and empty holes constitute "locally inconsistent interpretations". The second one is relevant as it constitutes the main reasoning step of the tableau algorithm proposed in the next section.

Property (9) is an example of a DTB where inconsistency does not propagate. Let $\mathfrak{T}_{12} = \langle \mathcal{T}_1, \mathcal{T}_2, \mathfrak{B}_{12} \rangle$ be a distributed T-box, in which $\mathcal{T}_1$ is inconsistent, i.e. $\mathcal{T}_1 \models \top \sqsubseteq \bot$, $\mathcal{T}_2$ does not contain any axioms, and $\mathfrak{B}_{12}$ contains the bridge rule $1 : A \xrightarrow{\sqsupseteq} 2 : G$. Let $\mathcal{I}_2$ be an interpretation of $\mathcal{T}_2$ such that it satisfies $G \sqsubseteq H$ and $H^{\mathcal{I}_2} = \emptyset$. The distributed interpretation $\left\langle \mathcal{I}_1^{\emptyset}, \mathcal{I}_2, r_{12} = \emptyset \right\rangle$ satisfies $\mathfrak{T}_{12}$. If, instead $G^{\mathcal{I}_2} \neq \emptyset$, then $\left\langle \mathcal{I}_1^{\emptyset}, \mathcal{I}_2, r_{12} = \emptyset \right\rangle$ does not satisfy the bridge rule $1 : A \xrightarrow{\sqsupseteq} 2 : G$. Whereas the distributed interpretation $\left\langle \mathcal{I}_1^{\Delta}, \mathcal{I}_2, r_{12} = \Delta_1 \times G^{\mathcal{I}_2} \right\rangle$ does. In both cases $\mathcal{I}_2$ is a "consistent interpretation" that does not satisfy $\top \sqsubseteq \bot$; this implies that $\mathfrak{T}_{12} \not\models 2 : \top \sqsubseteq \bot$.

To prove property (11) we have to show that, for any distributed interpretation $\mathfrak{I}$ that satisfies $\mathfrak{B}_{ij}$, if $A^{\mathcal{I}_i} \subseteq (\bigsqcup_{k=1}^{n} B_k)^{\mathcal{I}_i}$, then $G^{\mathcal{I}_j} \subseteq (\bigsqcup_{k=1}^{n} H_k)^{\mathcal{I}_j}$. Indeed, $G^{\mathcal{I}_j} \subseteq r_{ij}(A_i^{\mathcal{I}}) \subseteq r_{ij}\left(\bigcup_{k=1}^{n} B_k^{\mathcal{I}_i}\right) = \bigcup_{k=1}^{n} r_{ij}(B_k^{\mathcal{I}_i}) \subseteq \bigcup_{k=1}^{n} H_k^{\mathcal{I}_j} = (\bigsqcup_{k=1}^{n} H_k)^{\mathcal{I}_j}$.

Example 2. In the hierarchy WNP of the previous example there is no subsumption relation between Engineer and Worker. From bridge rules (1–3) and from the fact that in the ISCO-88 ontology the concept Architects_Engineers_And_Related_Professionals is a subclass of Professionals, it is impossible to infer that Engineers is a subclass of Worker, i.e. that in WNP Engineers $\sqsubseteq$ Worker. Similarly, the bridge rules (4) and (5) allow to infer that WNP classes Gatekeeper and Child are disjoint, i.e. that WNP : Gatekeeper $\sqcap$ Child $\sqsubseteq \bot$.

3 Reasoning in DDL

The main objective of this section is to investigate a decision procedure that computes subsumptions in DDL, i.e. if $\mathfrak{T} \models i : A \sqsubseteq B$.

A first proposal, described in [2], is based on reducing a DTB $\mathfrak{T}$ to an equivalent global T-box $\mathcal{T}_G$. Roughly speaking, this transformation indexes every atomic concept of the i-th T-box with the index i, and maps a complex concept, such as $C \sqcap D$ of $\mathcal{T}_i$, into $C_i \sqcap D_i$. Local domains are modeled by introducing a special concept $\top_i$ for each i. $\mathcal{T}_G$ contains transformations of the axioms of each $\mathcal{T}_i$ and the bridge rules, which are transformed into subsumptions by introducing a special role R_{ij} for each domain relation r_{ij}.

This approach, however, has three main limitations. The first one is theoretical, the second one is computational, and the third one is organizational.

From the theoretical point of view one has to notice that not all distributed interpretations can be mapped into the interpretation of a global T-box $\mathcal{T}_G$. In particular, distributed interpretations that contain holes cannot be represented by any global interpretation. This means that the reformulation in a global ontology can be used only when the DTB is consistent for every i, i.e. for every $i \in I$, $\mathfrak{T} \not\models i : \top \sqsubseteq \bot$.

From the computational point of view, it is widely agreed that reasoning in a structured set of local modules that communicate with each other is more efficient than reasoning in a global unstructured model. More formal argumentation on the advantages of partitioned reasoning can be found in [1, 17].

Finally, from the organizational perspective, the global theory approach requires access to the whole ontology, while in many practical cases ontologies can be accessed only via a query interface. Therefore, ontologies can be seen as black-boxes merely providing the possibility to check whether they satisfy a certain subsumption.

To overcome the mentioned above problems, our proposal is to build a distributed decision procedure, based on a *distributed tableau* method, which is defined on top of the *local tableaux procedures*, i.e. a procedures that build tableaux in local ontologies. We suppose to have a set of procedures $\mathbf{Tab}_i$, one for each $i \in I$. Given a concept C, $\mathbf{Tab}_i$ returns a *tableau* for C in $\mathcal{T}_i$. We use the notion of $\mathcal{SHIQ}$-tableaux defined in [11].

In order to get the intuition of the algorithm we make some simplifying assumptions. We will consider the case of only two ontologies (T-boxes), and unidirectional mappings (bridge rules) between them. Formally, $\mathfrak{T}_{12} = \langle \mathcal{T}_1, \mathcal{T}_2, \mathfrak{B}_{12} \rangle$. Later on, we relax these assumptions and extend our results to the general case.

Example 3. Suppose that $\mathcal{T}_1$ contains axioms $A_1 \sqsubseteq B_1$ and $A_2 \sqsubseteq B_2$, $\mathcal{T}_2$ does not contain any axiom, and that $\mathfrak{B}_{12}$ contains the following bridge rules:

$$1 : B_1 \xrightarrow{\sqsubseteq} 2 : H_1 \qquad 1 : B_2 \xrightarrow{\sqsubseteq} 2 : H_2 \tag{12}$$

$$1 : A_1 \xrightarrow{\sqsupseteq} 2 : G_1 \qquad 1 : A_2 \xrightarrow{\sqsupseteq} 2 : G_2 \tag{13}$$

Let us show that $\mathfrak{T}_{12} \models_d 2 : G_1 \sqcap G_2 \sqsubseteq H_1 \sqcap H_2$, i.e. that for any distributed interpretation $\mathfrak{I} = \langle \mathcal{I}_1, \mathcal{I}_2, r_{12} \rangle$, $(G_1 \sqcap G_2)^{\mathcal{I}_2} \subseteq (H_1 \sqcap H_2)^{\mathcal{I}_2}$.

1. Suppose that by contradiction there is an $x \in \Delta_2$ such that $x \in (G_1 \sqcap G_2)^{\mathcal{I}_2}$ and $x \notin (H_1 \sqcap H_2)^{\mathcal{I}_2}$.
2. Then $x \in G_1^{\mathcal{I}_2}$, $x \in G_2^{\mathcal{I}_2}$, and either $x \notin H_1^{\mathcal{I}_2}$ or $x \notin H_2^{\mathcal{I}_2}$.
3. Let us consider the case where $x \notin H_1^{\mathcal{I}_2}$. From the fact that $x \in G_1^{\mathcal{I}_2}$, by the bridge rule (13), there is $y \in \Delta_1$ with $\langle y, x \rangle \in r_{12}$, such that $y \in A_1^{\mathcal{I}_1}$.
4. From the fact that $x \notin H_1^{\mathcal{I}_1}$, by bridge rule (12), we can infer that for all $y \in \Delta_1$ if $\langle y, x \rangle \in r_{12}$ then $y \notin B_1^{\mathcal{I}_1}$.
5. But, since $A \sqsubseteq B \in \mathcal{T}_1$, then $y \in B_1^{\mathcal{I}_1}$, and this is a contradiction.
6. The case where $x \notin H_2^{\mathcal{I}_2}$ is analogous.

The above reasoning can be seen as a combination of a tableau in $\mathcal{T}_2$ with a tableau in $\mathcal{T}_1$, as it shown in Figure 3.

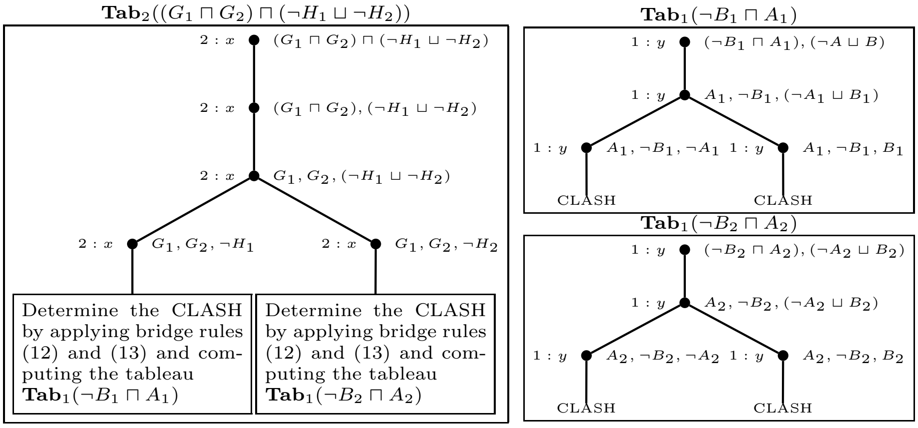

Fig. 3. An example of distributed tableau.

Let us formalize the above intuitions.

Definition 7. *Given a set of bridge rules $\mathfrak{B}_{12}$ from $\mathcal{DL}_1$ to $\mathcal{DL}_2$, the operator $\mathfrak{B}_{12}(.)$, taking as input a T-box in $\mathcal{DL}_1$ and producing a T-box in $\mathcal{DL}_2$, is defined as follows:*

$$\mathfrak{B}_{12}(\mathcal{T}_1) = \left\{ G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \;\middle|\; \begin{array}{l} \mathcal{T}_1 \models A \sqsubseteq \bigsqcup_{k=1}^{n} B_k, \\ 1 : A \xrightarrow{\sqsupseteq} 2 : G \in \mathfrak{B}_{12}, \\ 1 : B_k \xrightarrow{\sqsubseteq} 2 : H_k \in \mathfrak{B}_{12}, \textit{ for } 1 \leq k \leq n \end{array} \right\}$$

Lemma 1. *Let $\mathfrak{T}_{12} = \langle \mathcal{T}_1, \mathcal{T}_2, \mathfrak{B}_{12} \rangle$ be a distributed T-box, then:*

$$\mathfrak{T}_{12} \models 2 : X \sqsubseteq Y \iff \mathcal{T}_2 \cup \mathfrak{B}_{12}(\mathcal{T}_1) \models X \sqsubseteq Y \tag{14}$$

To prove this lemma, we need to introduce the definition of a *disjoint union of interpretations*. Given two interpretations $\mathcal{I} = \left\langle \Delta_{\mathcal{I}}, (.)^{\mathcal{I}} \right\rangle$ and $\mathcal{J} = \left\langle \Delta_{\mathcal{J}}, (.)^{\mathcal{J}} \right\rangle$, which are not holes, the *disjoint union of* $\mathcal{I}$ *and* $\mathcal{J}$, denoted by $\mathcal{I} \uplus \mathcal{J}$, is defined as $\left\langle \Delta_{\mathcal{I} \uplus \mathcal{J}}, (.)^{\mathcal{I} \uplus \mathcal{J}} \right\rangle$, where $\Delta_{\mathcal{I} \uplus \mathcal{J}} = \Delta_{\mathcal{I}} \uplus \Delta_j$ and $(.)^{\mathcal{I} \uplus \mathcal{J}}$ is defined as follows:

1. for any atomic concept A, $(A)^{\mathcal{I} \uplus \mathcal{J}} = A^{\mathcal{I}} \uplus A^{\mathcal{J}}$;
2. for any primitive role R, $R^{\mathcal{I} \uplus \mathcal{J}} = R^{\mathcal{I}} \uplus R^{\mathcal{J}}$;
3. for any constant a, $a^{\mathcal{I} \uplus \mathcal{J}} = a^{\mathcal{I}}$.

Disjoint union of interpretations preserves the interpretation of concepts, i.e. for any concept A, $A^{\mathcal{I} \uplus \mathcal{J}} = A^{\mathcal{I}} \uplus A^{\mathcal{J}}$, and therefore
$(\mathcal{I} \uplus \mathcal{J} \models A \sqsubseteq B)$ if and only if $(\mathcal{I} \models A \sqsubseteq B$ and $\mathcal{J} \models A \sqsubseteq B)$.

Lemma 2. *If $\mathcal{T} \not\models A \sqsubseteq B$ and $\mathcal{T} \not\models C \sqsubseteq D$, then there is an interpretation $\mathcal{I}^*$ of $\mathcal{T}$ such that $\mathcal{I}^* \not\models A \sqsubseteq B$ and $\mathcal{I}^* \not\models C \sqsubseteq D$.*

Proof (of Lemma 2). Let $\mathcal{I}$ and $\mathcal{J}$ be two interpretations of $\mathcal{T}$ such that $\mathcal{I} \not\models A \sqsubseteq B$ and $\mathcal{J} \not\models C \sqsubseteq D$. By induction on the structure of the complex concepts X and Y, we can prove that

$$(\mathcal{I} \uplus \mathcal{J} \models X \sqsubseteq Y) \text{ if and only if } (\mathcal{I} \models X \sqsubseteq Y \text{ and } \mathcal{J} \models X \sqsubseteq Y) \tag{15}$$

Let $\mathcal{I}^*$ be $\mathcal{I} \uplus \mathcal{J}$. By property (15) we have that $\mathcal{I}^* \models \mathcal{T}$, $\mathcal{I}^* \not\models A \sqsubseteq B$ and $\mathcal{I}^* \not\models C \sqsubseteq D$.

Proof (of Lemma 1). The $\Longleftarrow$ direction coincides with property (11), which has been already proved. Let us prove the $\Longrightarrow$ direction.

Consider that $1 : A \xrightarrow{\sqsupseteq} 2 : G \in \mathfrak{B}_{12}$ and $1 : B_k \xrightarrow{\sqsubseteq} 2 : H_k \in \mathfrak{B}_{12}$, for $1 \leq k \leq n$.

Let $\mathcal{I}_2$ be an interpretation of $\mathcal{T}_2 \cup \mathfrak{B}_{12}(\mathcal{T}_1)$ such that $\mathcal{I}_2 \not\models X \sqsubseteq Y$. For any concept G and any $d \in G^{\mathcal{I}_2}$, let us distinguish two cases:

1) If $\mathcal{T}_1 \models A \sqsubseteq \bot$, then let $\mathcal{I}_1^d$ be a hole, and thus the domain relation is $r_{12} = \Delta_{\mathcal{I}_1} \times G^{\mathcal{I}_2}$.

2) If $\mathcal{T}_1 \not\models A \sqsubseteq \bot$, then let $H_1, \dots H_n \in \mathbf{H}$ such that $d \not\in H_k$, for $0 \leq k \leq n$. This means that $\mathcal{T}_2 \cup \mathfrak{B}_{12}(\mathcal{T}_1) \not\models G \sqsubseteq \bigsqcup_{k=1}^n H_k$, and therefore $G \sqsubseteq \bigsqcup_{k=1}^n H_k \not\in \mathfrak{B}_{12}(\mathcal{T}_1)$. This implies that $\mathcal{T}_1 \not\models A \sqsubseteq \bigsqcup_{k=1}^n B_k$. Let $\mathcal{I}_1^d$ be a model of $\mathcal{T}_1$ such that $\mathcal{I}_1^d \not\models A_i \sqsubseteq \bigsqcup_{k=1}^n B_k$, and let $d' \in A^{\mathcal{I}_1^d}$ such that $d' \not\in (\bigsqcup_{k=1}^n B_k)^{\mathcal{I}_1^d}$. Let us add $\langle d', d \rangle$ to r_{12}.

Taking $\mathcal{I}_1 = \biguplus_{d \in \Delta_2} \mathcal{I}_1^d$ (the disjoint union of the models $\mathcal{I}_1^d$), let us prove that $\langle \mathcal{I}_1, \mathcal{I}_2, r_{12} \rangle$ is a model for the local T-boxes. Indeed, by definition $\mathcal{I}_2 \models \mathcal{T}_2$, and since for each $d \in \Delta_2$ and each $A \sqsubseteq B \in \mathcal{T}_1$, $\mathcal{I}_1^d \models A \sqsubseteq B$, we have that $\mathcal{I}_1 = \biguplus_{d \in \Delta_2} \mathcal{I}_1^d \models A \sqsubseteq B$.

Finally, let us prove that $\langle \mathcal{I}_1, \mathcal{I}_2, r_{12} \rangle$ satisfies also the bridge rules in $\mathfrak{B}_{12}$:

Onto bridge rule $1 : A \xrightarrow{\sqsupseteq} 2 : G$. We have to prove that $r_{12}(A^{\mathcal{I}_1}) \supseteq G^{\mathcal{I}_2}$.

Let $d \in G^{\mathcal{I}_2}$. Then by construction there is a $d' \in \Delta_{\mathcal{I}_1}$, such that $d' \in A^{\mathcal{I}_1}$.

Into bridge rule $1 : B \xrightarrow{\sqsubseteq} 2 : H$. We have to prove that $r_{12}(B^{\mathcal{I}_1}) \subseteq H^{\mathcal{I}_2}$.

Let $d' \in B^{\mathcal{I}_1}$, and $\langle d', d \rangle \in r_{12}$. Let us consider the following two cases:

1. If $\langle d', d \rangle$ has been added to r_{12} because of the bridge rule $1 : A \xrightarrow{\sqsupseteq} 2 : G$ with $\mathcal{T}_1 \models A \sqsubseteq \bot$, then we have that $d \in G^{\mathcal{I}_2}$ and since $\mathcal{I}_2 \models G \sqsubseteq H$ for any $H \in \mathbf{H}$, we have that $d \in H^{\mathcal{I}_2}$.
2. If $\langle d', d \rangle$ has been added to r_{12} because of the bridge rule $1 : A \xrightarrow{\sqsupseteq} 2 : G$ with $\mathcal{T}_1 \not\models A \sqsubseteq \bot$, then there is an interpretation $\mathcal{I}_1^d$, for some $d \in \Delta_2$ such that $d' \in B^{\mathcal{I}_1^d}$. Let us prove that $d \in H^{\mathcal{I}_2}$. By construction we have that H is different from all the $H_1, \dots, H_n$ such that $d \not\in H_1^{\mathcal{I}_2}, \dots, d \not\in H_n^{\mathcal{I}_2}$, and therefore $d \in H^{\mathcal{I}_2}$.

Lemma 1 provides the basic property that can be applied in order to settle the subsumption problem in DDL by means of a fixpoint application of the $\mathfrak{B}_{ij}$ operators.

Definition 8. *Let* $\mathfrak{T} = \langle \mathbf{T}, \mathfrak{B} \rangle$ *be a distributed T-box. Then:*
$\mathfrak{B}(\mathbf{T}) = \{\mathcal{T}_i \cup \bigcup_{j \neq i} \mathfrak{B}_{ij}(\mathcal{T}_i)\}_{i \in I}$.
$\mathfrak{B}^*(\mathbf{T})$ *is the smallest fix-point of* $\mathfrak{B}$ *containing* $\mathbf{T}$.

Theorem 1 (Soundness and completeness). *Given a distributed T-box* $\mathfrak{T} = \langle \mathbf{T}, \mathfrak{B} \rangle$, $\mathfrak{T} \models i : X \sqsubseteq Y$ *if and only if* $\mathfrak{B}^*(\mathbf{T})_i \models X \sqsubseteq Y$.

Theorem 1 provides the theoretical support for defining a distributed decision procedure for a subsumption in DDL. Such procedure can be implemented by combination of local reasoning, i.e. checking the fact that $\mathcal{T}_i \models A \sqsubseteq B$, with the $\mathfrak{B}$ propagation operator.

Similarly to description logics reduction of subsumption to unsatisfiability, we rephrase the problem of deciding $\mathfrak{T} \models i : A \sqsubseteq B$ into the problem of not finding a distributed interpretation $\mathfrak{I}$ of $\mathfrak{T}$, with $\mathcal{I}_i$ different from a hole and such that $(A \sqcap \neg B)^{\mathcal{I}_i} \neq \emptyset$.

Algorithm 1 defines a distributed procedure $\mathbf{dTab}_i$ for each $i \in I$. It takes as input a complex concept Φ to be verified and returns (un)satisfiability. The algorithm first checks for the loops, then builds a local completion tree T by running local tableau algorithm **Tab**, and further attempts to close open branches in tree checking bridge rules, which are capable of producing clash in nodes of T. According to the local tableau algorithm, each node x introduced during creation of the completion tree is labeled with a function $L(x)$ containing concepts that x must satisfy.

4 Prototype implementation

To evaluate the proposed distributed reasoning procedure we built a prototype intended to model P2P architecture given in Figure 1. The prototype deals with ontologies specified in OWL [10] and mappings described in C-OWL [3]. Ontologies and mappings are collected and managed by peer ontology managers, which are responsible for providing reasoning services on local ontologies. Among such services are checking consistency, performing classification, checking entailment, and others. The peer ontology manager is implemented as a java application, that can be deployed on a web server and accessed via HTTP.

The key role in the ontology manager is played by a distributed reasoning engine, implementing developed distributed tableau algorithm. The kernel of the engine is formed by Pellet[6]. Pellet is an open source java implementation of OWL DL reasoner, based on the use of tableaux algorithms developed for expressive description logics. Extension of the core reasoning functionality of Pellet transforms it to its distributed successor called *D-Pellet*.

To depicture the life cycle of D-Pellet, consider the case where a peer ontology manager is asked to perform one of the supported reasoning services in the particular local ontology it maintains. The ontology manager submits this query to D-Pellet, which in turn invokes the relative core Pellet functionality and checks

[6] `http://www.mindswap.org/2003/pellet`

Algorithm 1 Distributed reasoning procedure

dTab$_j(\Phi)$

BEGIN
If this procedure has been already called with the parameter Φ then return satisfiable.
T=**Tab**$_j(\Phi)$; {perform local reasoning and create completion tree}
if (T is not clashed) **then**
  for each open branch β in T **do**
    repeat
      select node $x \in \beta$ and an $i \neq j$;
      $\mathbb{C}_i^{onto}(x) = \{C \mid i : C \xrightarrow{\sqsupseteq} j : D,\ D \in L(x)\}$;
      $\mathbb{C}_i^{into}(x) = \{C \mid i : C \xrightarrow{\sqsubseteq} j : D, \neg D \in L(x)\}$;
      if $((\mathbb{C}_i^{onto}(x) \neq \emptyset)$ and $(\mathbb{C}_i^{into}(x) \neq \emptyset))$ **then**
        for each $C \in \mathbb{C}^{onto}$ **do**
          if (**dTab**$_i(C \sqcap \neg \bigsqcup \mathbb{C}_i^{into})$ is not satisfiable) **then**
            close β; {clash in x}
            break; {verify next branch}
          end if
        end for
      end if
    until ((β is open) and (there exist not verified nodes in β))
  end for{all branches are verified}
end if
if (T is clashed) **then**
  return unsatisfiable;
else
  return satisfiable;
end if
END

for available mappings. Mappings can generate subqueries which are dispatched by the ontology manager to the corresponding foreign ontology manager. In turn, this starts another reasoning cycle. The reasoning stops when the initial D-Pellet receives the answers to the subproblems it sent out. Analysis of the subproblem answers defines the final reasoning result.

To facilitate the use of peer ontology managers, i.e. developing and assigning to them local ontologies and mappings, as well as to evaluate the reasoning capabilities of D-Pellet, we extended the Protégé-2000 ontology editing tool [16] with an *ontology manager plug-in.* The plug-in enriches core Protégé-2000 , with the possibility develop an ontology related via semantic mappings with other remote ontologies. Plug-in supports: (i) manual development of mappings between pairs of OWL ontologies (integration with some automatic mapping generation tool is considered in the future), (ii) publishing the developing ontology and related to it mappings to the peer ontology manager, (iii) requesting local/global classification and check of consistency services of the associated peer ontology manager, allowing to see how established mappings affects the developing ontology.

The described Protégé-2000 ontology manager plug-in can be of particular benefit during development of modular ontologies. Here by modular ontology we assume the ontology that comprises a set of autonomous modules, which are interrelated via semantic mappings. The important feature of the plug-in is its ability to preserve full autonomy of module, including reasoning in it.

5 Related work

From a theoretical perspective, presented work is an extension of the results introduced in [2]. The main contributions of this paper are: (i) the distributed tableau algorithm, which constitutes the first attempt for building a sound and complete distributed decision procedure for a set of interconnected ontologies, and (ii) implementation of the proposed algorithm in a peer-to-peer architecture.

In [12], it has been shown that DDL can be represented in a much richer theoretical framework for integrating different logics, called $\mathcal{E}$-connections. $\mathcal{E}$-connections allow to state relations between a set of logical frameworks using *binary link relations*. Bridge rules can be seen as a special case of binary link relations. The embedding, described in [12], holds in the case when local ontologies of distributed T-box are not interpreted in holes. In this limited case, the embedding allows to state that the complexity class of the satisfiability problem of DDL as a function of the complexity classes of the local satisfiability problems. In this paper, we introduced a distributed tableau for DDL capable of dealing with holes.

DDL inherited a lot of ideas form the other logics for distributed systems. It is a subclass of Multi Context Systems (MCS), the general framework for contextual reasoning [6] developed in Trento's group. Yet DDL is an extension of propositional MCS [7, 4, 18]. The satisfiability problem in propositional MCS is described in [19, 17]. And finally, DDL is a special case of Distributed First Order Logics (DFOL) described in [5].

From the practical point of view, the remarkable feature of proposed in this paper D-Pellet system and its integration into Protégé-2000 , with respect to the existing multiple ontology management systems (such as Hozo [21], KAON [13], PROMPT [15]), is the support of distributed reasoning organized in a peer-to-peer architecture. D-Pellet system is comparable with RICE[7] (RACER Interactive Client Environment). RICE provides a graphical interface to RACER, using the RACER API, and allows to browse T-boxes and A-boxes, plus provides various querying facilities. RICE supports multiple T-boxes and A-boxes, but comparing to D-Pellet's integration in Protégé-2000 , it does not support mappings between ontologies, and peer-to-peer interaction between ontologies.

6 Conclusions

In this paper we have presented a tableau-based distributed reasoning procedure for Distributed Description Logics. This procedure constitutes a method

[7] `http://www.b1g-systems.com/ronald/rice/`

for combining existing tableaux reasoning procedures for stand alone Description Logics.

We also described the implemented prototype of java-based peer ontology manager implementing developed reasoning procedure. To facilitate the creation of semantic mappings between OWL ontologies and provide a mapping-aware ontology consistency checker and classifier we built an ontology manager plug-in for Protégé-2000 ontology development platform enabling joint use of these two applications.

References

1. E. Amir and S. McIlraith. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence*, 2003. Accepted for publication.
2. A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, (1):153–184, 2003.
3. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-owl: Contextualizing ontologies. In *Proceedings of the 2d International Semantic Web Conference (ISWC2003)*, pages 164–179, 2003.
4. C. Ghidini and F. Giunchiglia. Local model semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.
5. C. Ghidini and L. Serafini. Distributed first order logics. In *Proceedings of the Frontiers of Combining Systems*, pages 121–139, 2000.
6. F. Giunchiglia. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993.
7. F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics (or: How we can do without modal logics). *Artificial Intelligence*, 65(1):29–70, 1994.
8. V. Haarslev and R. Moller. Racer system description. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR2001)*, pages 701–706, 2001.
9. I. Horrocks and P. F. Patel-Schneider. Fact and dlp. In *Proceedings of the Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX1998)*, pages 27–30, 1998.
10. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
11. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of IGPL*, 8(3):239–263, 2000.
12. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence*, 2004. To appear.
13. A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies in the semantic web. *The International Journal on Very Large Data Bases (VLDB)*, 12(4):286–302, 2003.
14. M.Bonifacio, P.Bouquet, and P.Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.
15. N. F. Noy and M. A. Mussen. The prompt suite : interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 56(6):983–1024, 2003.

16. N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.
17. F. Roelofsen and L. Serafini. Complexity of contextual reasoning. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI2004)*, 2004. Accepted for publication.
18. L. Serafini and F. Giunchiglia. Ml systems: A proof theory for contexts. *Journal of Logic, Language and Information*, 11(4):471–518, 2002.
19. L. Serafini and F. Roelofsen. Satisfiability for propositional contexts. In *Proceedings of the Principles of Knowledge Representation and Reasoning (KR2004)*, 2004. Accepted for publication.
20. H. Stuckenschmidt and M. Klein. Integrity and change in modular ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2003)*, 2003.
21. E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. An environment for distributed ontology development based on dependency management. In *Proceedings of the 2nd International Semantic Web Conference (ISWC2003)*, pages 453–468, 2003.