Distributed Instance Retrieval in Heterogeneous Ontologies

Luciano Serafini[1] and Andrei Tamilin[1,2]

[1] ITC-IRST, Via Sommarive 18, 38050 Trento, Italy
luciano.serafini@itc.it
[2] DIT - University of Trento, Via Sommarive 14, 38050 Trento, Italy
andrei.tamilin@dit.unitn.it

Abstract. The problem of ontology querying is of a crucial importance for a practically usable semantic web. Recent research in standard Description Logics has produced a number of algorithms and implementations for reasoning with instances for answering ontological queries. In this work, we extend these techniques to enable a querying of distributed heterogeneous ontologies with heterogeneous populations. We follow a peer-to-peer paradigm that of combining local ontology query answering with answering over other semantically related ontologies. Our approach is formally grounded on Distributed Description Logics (DDLs) and limited to the instance retrieval queries. The contribution of the paper is twofold. From the theoretical perspective we extend the semantics of DDLs with instances which allows to define an instance retrieval algorithm. From the practical hand, we describe how the proposed algorithm can be implemented on top of DRAGO distributed reasoner.

1 Introduction

The semantic web steadily evolves growing into a container of multiple autonomously developed, distributed and heterogeneous ontologies. These ontologies are differently populated with volumes of instances. In order to provide a coherent access to such a web of ontologies, one needs to tackle with a *heterogeneity* problem arising on both *conceptual* and *instance* levels.

Due to the service-oriented architecture of a web, the access to web of ontologies is organized via set of ontological services. We will distinguish between two paradigmatic ontological services, namely *reasoning services* for dealing with a conceptual part of ontology and *querying services* for manipulating ontology populations. An example of fundamental reasoning service is checking a concepts subsumption, i.e., whether one concept is more/less general than the other concept. In turn, an important querying service is an *instance retrieval*, i.e., finding all instances of a particular concept. Both groups of services play a crucial role for building a practically usable semantic web.

In our previous work [2] we have addressed the problem of providing reasoning services over heterogeneous ontologies without considering their instances. The conceptual heterogeneity in that approach was resolved by defining semantic relations between heterogeneous concepts belonging to different ontologies, called a semantic mapping. The reasoning is then accomplished through a combination via semantic mapping of standard local reasoning chunks, internally executed in each separate ontology. The technique is formally grounded on the framework of Distributed Description Logics (DDLs) [3] and the reasoning algorithm, called a distributed tableaux, is built as an extension to a standard Description Logics tableaux.

In this paper, we propose an extension to DDLs framework and make a step towards shedding the light on the second crucial group of ontological services, namely querying heterogeneous ontologies with heterogeneous populations. From the theoretical side, we extend the semantics of DDLs with an individual-level mapping, which allows us deal with the heterogeneity at the instance level. Afterwards, we propose a distributed instance retrieval algorithm, which is based on a peer-to-peer paradigm that of combining local ontology instance retrieval with instance retrieval over other semantically related ontologies. From the practical side, we describe how the proposed algorithm can be implemented in DRAGO distributed reasoning system [2].

The proposed distributed instance retrieval algorithm composes standard Description Logics (DLs) querying algorithms [9, 6–8, 10, 14]. Practically efficient implementations of query answering over A-boxes weaken supported DL languages posing certain restrictions on the structure of A-boxes [15, 16]. Such weakening, not only increases the efficiency, but also simplifies the answering algorithm itself. For the sake of simplicity, in this paper we give preference to such simplified versions of DL answering algorithms.

The paper is structured as follows. In Section 2 we start with a simple motivating scenario to depict a problem. Section 3 reviews the basics concerning dealing with instances in standard Description Logics. Then, Section 4 discusses representing assertional part of Distributed Description Logics. Section 5 describes the operational semantics of bridge rules and individual correspondences, and further, in Section 6, we apply that semantics to the problem of distributed instance retrieval. Section 7 finalizes the paper with the description of how DRAGO reasoner is extended to support the proposed instance retrieval algorithm.

2 A motivating scenario

Consider this simple scenario. The human resource offices of a university and one of its department utilize ontologies for describing research staff. The university exploits ontology O^{Uni}, while the department uses O^{Dept}. The obvious overlap of the domains of interest in both ontologies occurs since every person that works in the department belongs to the university. However, let us additionally suppose that O^{Uni} instantiates its concept $PersonContact$ in the domain of personal e-mails, while O^{Dept}, respectively, instantiates its concept $PersonContact$ in the domain of world wide web personal homepages.

When the ontology of the university is asked for all personal contacts, to guarantee a complete answer, this query should be propagated to the departmental ontology expanding the result set. However, since instantiations of semantically related concepts are expressed using different "formats", one needs a mechanism of semantic preserving instance transformation. In our example, when retrieving instances of $PersonContact$ in O^{Uni}, i.e., contacts of the university staff, besides locally found e-mails one should also augment the response by retrieving the instances of $PersonContact$ in O^{Dept}, i.e., personal homepages, and further project them into domain of O^{Dept}, i.e., "visit" every homepage found and extract, if any, personal e-mails from the pages.

3 Individuals in Description Logics

Let us briefly review the basic definitions and major results concerning reasoning with instances in the standard Description Logics (DL) [1].

A DL *knowledge base* $\mathcal{K}$ is constituted by two components, a terminological $\mathcal{T}$ (T-box) and an assertional $\mathcal{A}$ (A-box). Terminological statements are expressed in form of inclusion axioms, such as $C \sqsubseteq D$, $R \sqsubseteq S$, where C, D and R, S are concepts and roles respectively. The A-box asserts facts about individuals in form of concept memberships $C(a)$ and role memberships $R(a, b)$, where a, b are individuals.

The semantics of DL is given according to the notion of interpretation. An interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ consists of a nonempty set Δ (the domain) and an interpretation function $\cdot^{\mathcal{I}}$ that maps every concept to a subset of Δ, every role to a subset $\Delta \times \Delta$, and every individual to an element of Δ.

An interpretation $\mathcal{I}$ satisfies the terminological statement $C \sqsubseteq D$ ($R \sqsubseteq S$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$), and the assertional statement $C(a)$ ($R(a, b)$) iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). $\mathcal{I}$ satisfies (is a model for) the T-box $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) if it satisfies each assertion in $\mathcal{T}$, and $\mathcal{I}$ satisfies (is a model for) the A-box $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$) if it satisfies each assertion in $\mathcal{A}$. Similarly, $\mathcal{I}$ satisfies (is a model for) the knowledge base $\mathcal{K}$ (written $\mathcal{I} \models \mathcal{K}$) if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. A statement α is a logical consequence of (is entailed by) the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ (written $\mathcal{K} \models \alpha$) iff for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$, then $\mathcal{I} \models \alpha$.

Reasoning is the fundamental process of discovering facts entailed by knowledge base. Interesting inference problems for DLs include

- concept satisfiability: given a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a concept C is satisfiable w.r.t. $\mathcal{K}$ iff there exists a model $\mathcal{I}$ of $\mathcal{K}$ s.t. $C^{\mathcal{I}} \neq \emptyset$
- subsumption: a concept C is subsumed by concept D w.r.t. $\mathcal{K}$ iff $\mathcal{K} \models C \sqsubseteq D$
- classification: finding an explicit representation of the concept taxonomy entailed by the knowledge base
- instantiation: whether an individual a is an instance of a given concept C, i.e., $\mathcal{K} \models C(a)$
- retrieval: which individuals are instances of a given concept
- realization: what is the most specific named concept an individual is an instance of.

Historically, the majority of applications were mainly interested in Description Logics with only terminology (no A-box). In particular, pure terminological DLs were considered sufficient enough for certain ontology engineering tasks, reasoning with database schemas, and etc. The investigation of such DLs showed that due to the presence of tree model property, terminological reasoning can be performed by tableaux-based algorithms with good computational properties [5]. This research led to implementation of efficient, highly optimized state-of-the-art DL reasoners for the range of expressive DL languages, such as for example $\mathcal{SHIQ}$ [18, 17, 19].

The difficulty in extending the tableaux algorithms to the case of non-empty A-boxes is explained by the interactions between individuals and role operators. Such interactions can lead to the loss of tree model property and cause a bad computational behavior and poor performance of reasoning algorithms [12, 11]. The situation is getting even worse if a powerful generalization of A-box individuals, so called nominals (classes with a singleton extension), is allowed.

It was studied out that certain Description Logics enjoy an A-box partitioning property which allows to avoid bad interactions between individuals [13, 10]. The main idea behind such techniques is to eliminate role assertions from A-box by explicitating the consequences of such assertions. Once all role assertions are eliminated, individuals become independent from one another. This means that the original A-box can be partitioned into a set of separate A-boxes, each one describing the properties of a single individual. Such separation enables the reduction of a reasoning w.r.t. a knowledge base to a pure terminological reasoning. The completeness and correctness of this reduction depends on the expressivity of the language and particularly was proven to be applicable to $\mathcal{SHF}$ DL [13, 10].

Other techniques pose even more severe restrictions on the structure of an A-box for enabling reduction of reasoning to a terminological one. In particular, an Instance Store [15] utilizes so called role-free A-boxes. An A-box is said to be role-free if it contains concept membership assertions only. Such limitations allow to combine a terminological reasoner and a database with individuals for achieving a scalable query answering. Formally, the approach is sound and complete w.r.t. $\mathcal{ALCO}(\mathcal{D})$ knowledge bases. Despite this quite severe language restriction, the presented approach finds a wide application due to its scaling-up behavior.

4 Distributed Description Logics with Individuals

Distributed Description Logics (DDLs) is a framework designed to formalize the case of multiple ontologies interconnected by semantic mappings. In this section, we recall the definitions of DDLs as given in [3].

Given a set I of indeces enumerating ontologies, a DDL is a collection of description logics $\{\mathcal{D}L_i\}_{i \in I}$ corresponding to each ontology. We will denote a T-box of $\mathcal{D}L_i$ as $\mathcal{T}_i$ and an A-box as $\mathcal{A}_i$. The semantic mapping between ontologies are expressed as "bridge rules" between pairs of connects belonging to different ontologies.

A bridge rule from i to j is an expression of the following two forms:

$i : x \xrightarrow{\sqsubseteq} j : y$ – an *into-bridge rule*

$i : x \xrightarrow{\sqsupseteq} j : y$ – an *onto-bridge rule*

where x, y are concepts or individuals belonging to $\mathcal{DL}_i$ and $\mathcal{DL}_j$ respectively. The derived bridge rule $i : x \xrightarrow{\equiv} j : y$ can be defined as a conjunction of the corresponding into- and onto-bridge rule.

Despite this general form allowing for unrestricted mix of concepts with individuals, e.g., nominals, in this paper, we restrict the bridge rules to be expressions connecting only pairs of concepts. Therefore, bridge rules are allowed to connect only terminological parts of two DLs (ontologies). Moreover, we will prohibit the use of nominals in component logics $\mathcal{DL}_i$ of DDLs. This restriction is motivated by the hardness of dealing with nominals even in case of standard Description Logics (see Section 3 for more details).

Bridge rules from i to j express relations between i and j viewed from the *subjective* point of view of the j-th ontology. For example, the into-bridge rule $i : C \xrightarrow{\sqsubseteq} j : D$ intuitively says that, from the j-th point of view, the individuals in concept C in i correspond (via an approximation introduced by an implicit semantic domain relation) to a subset of the individuals in its local concept D.

A *distributed T-box (DTBox)* $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ 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 pairs of corresponding T-boxes.

In order to express correspondences between semantically related heterogeneous individuals between A-boxes, we follow the approach of introducing the individual-level equivalent of bridge rules:

$i : x \mapsto j : y$ – an *individual correspondence*

where x, y are individuals of $\mathcal{A}_i$ and $\mathcal{A}_j$ respectively.

Similarly to bridge rules, individual correspondences from i to j reflect a subjective point of view of the j-th ontology. For example, the individual correspondence $i : a \mapsto j : b$ intuitively expresses that according to j-th point of view, the individual b is one of the possible translations of the foreign individual a in the local domain of j. Notice, that this definition admits multiple translations of foreign individuals, i.e., one can simultaneously have $i : a \mapsto j : b$ and $i : a \mapsto j : b'$.

A *distributed A-box (DAbox)* $\mathfrak{A} = \langle \{\mathcal{A}_i\}_{i \in I}, \mathfrak{C} \rangle$ consists of a collection of A-boxes $\{\mathcal{A}_i\}_{i \in I}$ and a collection of individual correspondences $\mathfrak{C} = \{\mathfrak{C}_{ij}\}_{i \neq j \in I}$ between pairs of corresponding A-boxes.

A *distributed knowledge base (DKB)* $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$ is then a pair containing a distributed T-box and a distributed A-box.

The semantics of DDLs, graphically depicted on Figure 1, locally interprets each ontology by a standard Description Logics interpretation on its local domain. Since local domains Δ_i may be heterogeneous (e.g., see the motivation scenario in Section 2), the semantic correspondences between heterogeneous domains are modeled using so called domain relation r_{ij} from Δ_i to Δ_j and defined as a subset of $\Delta_i \times \Delta_j$. As we described in motivation scenario, if in Δ_1 and Δ_2 concept $PersonContact$ is populated in homepages and e-mails accordingly,

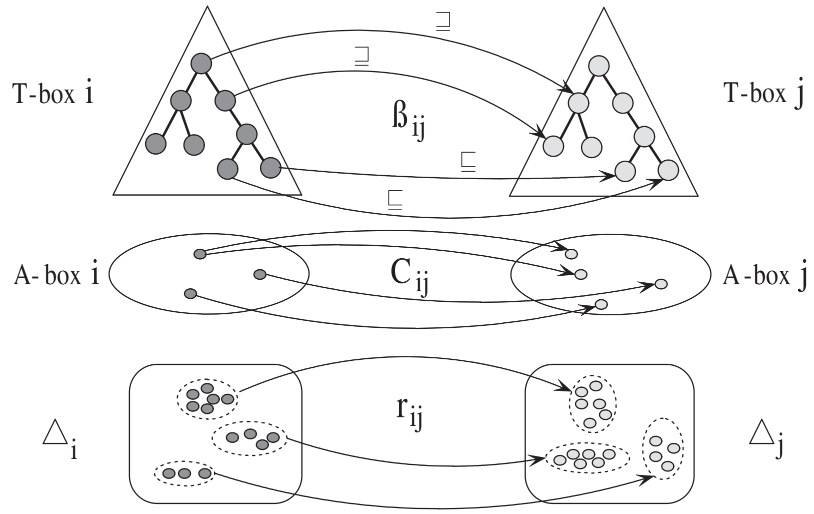


Fig. 1. Graphical intuition of Distributed Description Logics framework

then r_{12} could be function pseudo-coded as "visit the homepage and extract e-mails".

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

To deal with possible inconsistencies, DDLs utilizes the notion of a special non-classical interpretation $\mathcal{I}^{\epsilon}$, called a hole (see [4] for details), that interprets every concept or role in the empty set. The hole satisfies any A-box. To highlight that distributed interpretation $\mathfrak{I}$ can contain holes, we will subscript the inferences w.r.t. $\mathfrak{I}$ with a symbol "ϵ".

A distributed interpretation $\mathfrak{I}$ is said to satisfy (written $\mathfrak{I} \models_{\epsilon}$) the elements of distributed T-box $\mathfrak{T}$ if

- $\mathcal{I}_i \models_{\epsilon} A \sqsubseteq B$ for all $A \sqsubseteq B$ in $\mathcal{T}_i$
- $\mathfrak{I} \models_{\epsilon} i : A \xrightarrow{\sqsupseteq} j : G$, if $r_{ij}(A^{\mathcal{I}_i}) \supseteq G^{\mathcal{I}_j}$ (A, G are concepts of $\mathcal{T}_i, \mathcal{T}_j$)
- $\mathfrak{I} \models_{\epsilon} i : B \xrightarrow{\sqsubseteq} j : H$, if $r_{ij}(B^{\mathcal{I}_i}) \subseteq H^{\mathcal{I}_j}$ (B, H are concepts of $\mathcal{T}_i, \mathcal{T}_j$)
- $\mathfrak{I} \models_{\epsilon} \mathfrak{T}$, if for every $i, j \in I$, $\mathfrak{I} \models_{\epsilon} \mathcal{T}_i$ and $\mathfrak{I} \models_{\epsilon} \mathfrak{B}_{ij}$

Finally, $\mathfrak{T} \models_{\epsilon} i : C \sqsubseteq D$ if for every $\mathfrak{I}$, $\mathfrak{I} \models_{\epsilon} \mathfrak{T}$ implies $\mathfrak{I} \models_{\epsilon} i : C \sqsubseteq D$. We say that $\mathfrak{T}$ is *satisfiable* if there exists a $\mathfrak{I}$ such that $\mathfrak{I} \models_{\epsilon} \mathfrak{T}$. Concept $i : C$ is *satisfiable* with respect to $\mathfrak{T}$ if there is a $\mathfrak{I}$ such that $\mathfrak{I} \models_{\epsilon} \mathfrak{T}$ and $C^{\mathcal{I}_i} \neq \emptyset$.

Concerning the assertional part, a distributed interpretation $\mathfrak{I}$ is said to satisfy the elements of distributed A-box $\mathfrak{A}$ if

- $\mathcal{I}_i \models_{\epsilon} C(a)$, $\mathcal{I}_i \models_{\epsilon} R(a, b)$ for all assertions $C(a)$, $R(a, b)$ in $\mathcal{A}_i$
- $\mathfrak{I} \models_{\epsilon} i : x \mapsto j : y$, if $\langle x^{\mathcal{I}_i}, y^{\mathcal{I}_j} \rangle \in r_{ij}$

– $\mathfrak{I} \models_\epsilon \mathfrak{A}$, if for every $i, j \in I$, $\mathfrak{I} \models_\epsilon \mathcal{A}_i$ and $\mathfrak{I} \models_\epsilon \mathfrak{C}_{ij}$

Finally, $\mathfrak{A} \models_\epsilon i : C(a)$ ($\mathfrak{A} \models_\epsilon i : R(a,b)$) if for every $\mathfrak{I}$, $\mathfrak{I} \models_\epsilon \mathfrak{A}$ implies $\mathfrak{I} \models_\epsilon i : C(a)$ ($\mathfrak{I} \models_\epsilon i : R(a,b)$). We say that $\mathfrak{A}$ is *consistent* if there exists a $\mathfrak{I}$ such that $\mathfrak{I} \models_\epsilon \mathfrak{A}$.

A statement α is entailed by the distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$ (written $\mathfrak{K} \models_\epsilon$) iff for every distributed interpretation $\mathfrak{I}$, if $\mathfrak{I} \models_\epsilon \mathfrak{T}$ and $\mathfrak{I} \models_\epsilon \mathfrak{A}$, then $\mathfrak{K} \models_\epsilon \alpha$.

5 Operational semantics of bridge rules and individual correspondences

In this section, we analyze the semantics of bridge rules and individual correspondences in order to understand their operational behavior. In particular, we discuss semantical mechanisms allowing for propagating the terminological and assertional knowledge.

5.1 Terminological propagation

In our previous work [4], we characterized the distributed terminological inference using a fixed point bridge operator, which does a forward propagation of knowledge in DDLs. The basic idea preceding that result is that combination of onto- and into-bridge rules allows for directional propagating the terminological knowledge across ontologies in form of DL subsumption axioms.

Formally this reasoning pattern is formulated in the following proposition: Given a distributed T-box $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \{\mathfrak{B}_{ij}\}_{i \neq j \in I} \rangle$, if $\mathfrak{B}_{ij}$ contains $i : A \xrightarrow{\sqsupseteq} j : G$ and $i : B_k \xrightarrow{\sqsubseteq} j : H_k$ for $1 \leq k \leq n$ and $n \geq 0$, then:

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

where $\bigsqcup_{k=1}^{0} D_k$ denotes $\perp$. Figure 2 depicts this result for the case of $\mathfrak{T}_{12} = \langle \{\mathcal{T}_1, \mathcal{T}_2\}, \mathfrak{B}_{12} \rangle$ when a single pair of into-onto-bridge rules specified.

Taking the subsumption propagation pattern allowed to define a sound and complete distributed tableaux algorithm for a *distributed subsumption testing* in DDLs when its constituent DLs are expressed in (a subset of) $\mathcal{SHIQ}$ language [4]. Thus, in order to check whether $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ the standard autonomous $\mathcal{SHIQ}$-tableaux procedures for every member of distributed T-box were extended with an additional "bridge" expansion rule. The goal of that rule is to perform a lazy import (on demand) of propagated subsumption axioms into the nodes of standard tableaux so that the subsumption propagation is taken into account.

The characterization of knowledge propagation in [4] has been done by exploiting a disjoint union model property of DL. According to that property, given two models $\mathcal{I} = \left\langle \Delta_\mathcal{I}, \cdot^\mathcal{I} \right\rangle$ and $\mathcal{J} = \left\langle \Delta_\mathcal{J}, \cdot^\mathcal{J} \right\rangle$ of a $\mathcal{SHIQ}$ knowledge bases,

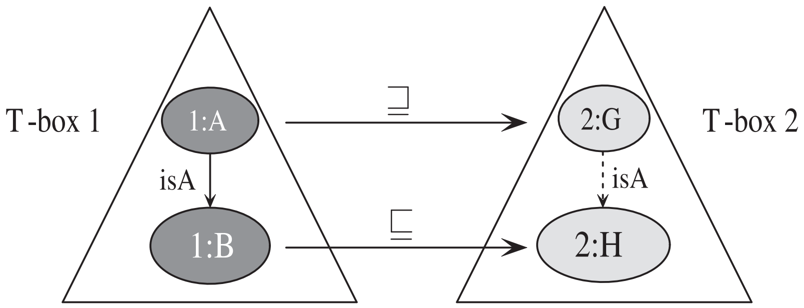
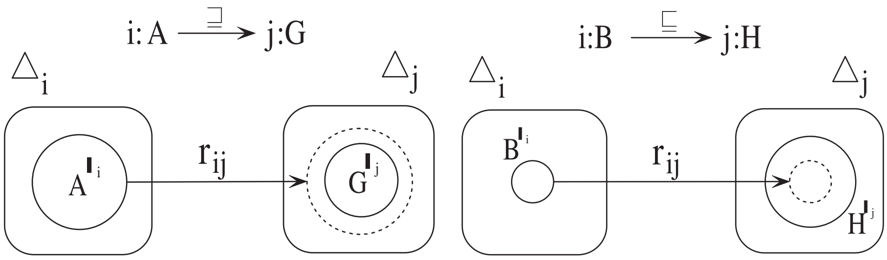


Fig. 2. Example of subsumption propagation in DDLs (inferred subsumption is dashed)

an interpretation $\mathcal{I} \uplus \mathcal{J} = \left\langle \Delta_{\mathcal{I} \uplus \mathcal{J}}, \cdot^{\mathcal{I} \uplus \mathcal{J}} \right\rangle$ is also a model. Unfortunately, as we pointed out in that work, not all DLs enjoy the disjoint union model property. Unrestricted use of nominals leads to the loss of that union property.

5.2 Assertional propagation

As we could see above, bridge rules constitute the channel for propagating knowledge in form of subsumption axioms across ontologies. Let us now investigate the consequence of bridge rules and individual correspondences in order to understand their affect on assertional part of knowledge in distributed knowledge base. We visualize on Figure 3 the semantics given to bridge rules in DDLs:



(a) Onto-bridge rule (b) Into-bridge rule

Fig. 3. Visualized semantics of bridge rules in DDLs

The special role of into-bridge rules can be intuitively grasped from these visualizations. Indeed, the semantics of into-bridge rule depicted on Figure 3(b) says: if in ontology i a concept B has an instance b then in ontology j there should exist such instance h of H that a pair $\left\langle b^{\mathcal{I}_i}, h^{\mathcal{I}_j} \right\rangle$ belongs to the domain relation r_{ij}.

Formally this intuition can be stated as an assertional propagation pattern: Given a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$, if $\mathfrak{B}_{ij}$ contains $i : B \xrightarrow{\sqsubseteq} j : H$

and $\mathfrak{C}_{ij}$ contains $i : b \mapsto j : h$, then:

$$\mathfrak{K} \models_\epsilon i : B(b) \Longrightarrow \mathfrak{K} \models_\epsilon j : H(h) \tag{2}$$

The result of applying the pattern (2) to DDLs describing two populated ontologies can be depicted graphically as given on Figure 4.

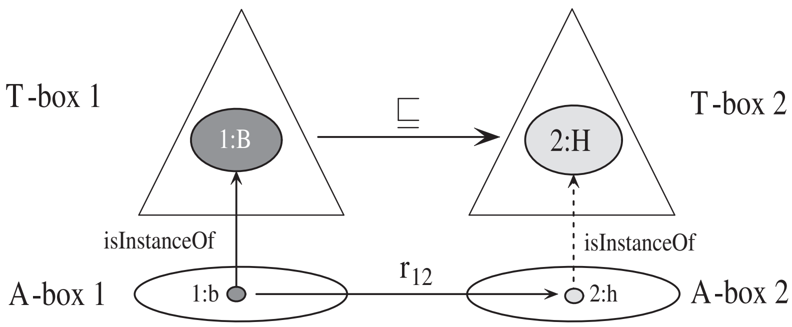


Fig. 4. Concept membership propagation in DDLs (inferred assertion is dashed)

According to the semantics of individual correspondence, $i : b \mapsto j : h$ means that the pair of individuals $\langle b^{\mathcal{I}_i}, h^{\mathcal{I}_j} \rangle$ belongs to the domain relation r_{ij}. On one hand, to have a complete set of individual correspondences a knowledge engineer should take all individuals x of ontology i, find all semantically corresponding individuals y in ontology j, and explicitly specify that $i : x \mapsto j : y$ (extensional approach). On the other hand, one can imagine a more compact definition, when an engineer specifies a transformation relation f_{ij} that allows to transform a set of individuals from i to individuals of j such that their interpretations respect the semantics of domain relation r_{ij}. In this later option, we can reformulate the pattern (2):

Given a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$ and a set of transformation relations f_{ij}, if $\mathfrak{B}_{ij}$ contains an into-bridge rule $i : B \xrightarrow{\sqsubseteq} j : H$, then:

$$\mathfrak{K} \models_\epsilon i : B(b) \Longrightarrow \mathfrak{K} \models_\epsilon j : H(f_{ij}(b)) \tag{3}$$

Consequently, we can give another reading to Figure 4. I.e., instance h was injected into A-box $\mathcal{A}_2$ with an assertion $2 : H(h)$ as a result of propagation via transformation of assertion $1 : B(b)$ from $\mathcal{A}_1$ along the into-bridge rule $i : B \xrightarrow{\sqsubseteq} j : H$.

It should be stressed again that a transformation relation f_{ij} plays a similar role on the level of individuals in A-boxes that the domain relation r_{ij} plays on the level of interpretation domains. However, since DL interpretation function maps individuals into elements of domain we can roughly think of the transformation relation as of a partial specification of the domain relation.

6 Distributed instance retrieval in DDLs

In this section, we focus on the combination of terminological and assertional propagation in DDLs in order to perform the task of distributed instance retrieval. Given a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$, a *distributed retrieval of instances* of a concept D in ontology i is a task of finding all individuals x that instantiate D, i.e., $\mathfrak{K} \models_\epsilon i : D(x)$.

Our proposal consists in building an instance retrieving mechanism for DDLs on top of standard DLs instance retrieval algorithms. We will describe the intuition of the algorithm for the case of distributed knowledge base with only two constituent ontologies, i.e., when $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle = \langle \langle \{\mathcal{T}_1, \mathcal{T}_2\}, \mathfrak{B}_{12} \rangle, \langle \{\mathcal{A}_1, \mathcal{A}_2\}, \mathfrak{C}_{12} \rangle \rangle$. Due to the direction of bridge rules from 1 to 2, we will denote ontology $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ as a *source* ontology, and $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle$ as a *target* ontology.

We will also put some restrictions on a structure of A-boxes $\mathcal{A}_1, \mathcal{A}_2$ admitting only those logics that enjoy the property of reduction reasoning with instances to a pure terminological reasoning (see Section 3 for more details). This will allow us to complete the part of local instance retrieval with sound and complete instance retrieval algorithms defined for these logics.

The terminological propagation results in refining a concept taxonomy of target ontology according to the pattern (1). Hereafter, we will refer to a taxonomy built with respect to the terminological propagation as a *distributed taxonomy* and a reasoner capable of that functionality as a *distributed terminological reasoner*.

The assertional propagation reveals in either (a) asserting new concept membership for existing individuals of target ontology when a pattern (2) is used, or (b) injecting new individuals to target ontology in accordance with pattern (3) and further asserting their concept membership similarly to (a). In both cases, we will refer to such concept membership assertions as *distributed concept membership*.

The consequence of both propagational aspects affects distributed instance retrieval. The resulting instances can be naturally subdivided into two groups: ones that were explicitly defined in the target ontology A-box, and the others that were injected as transformation of instances of source ontology. Therefore, the distributed retrieval problem should be approached in a 3-step manner: (1) retrieve all local instances with respect to the distributed taxonomy of the target ontology, (2) retrieve all relevant instances of the source ontology and apply transformation to them, and finally (3) merge the results of two previous steps.

Algorithmically, the process of distributed retrieving the instances of named concept D in ontology $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle$ w.r.t. a distributed knowledge base $\mathfrak{K}$ can be sketched as follows:

dRetrive$_2(D)$

1. compute a distributed taxonomy of $\mathcal{T}_2$
2. create a set $D_{\downarrow \mathcal{T}_2}$ of named concepts in $\mathcal{T}_2$ that are subsumed by a concept C w.r.t. a distributed T-box $\mathfrak{T}$ (these are equivalents and descendants of C in $\mathcal{T}_j$ computed w.r.t. $\mathfrak{T}$)

3. retrieve a set S^{local} of individuals in $\mathcal{A}_2$ using a standard DL instance retrieval algorithm with respect to the computed distributed taxonomy of $\mathcal{T}_2$
4. for every concept $H \in D_{\downarrow\mathcal{T}_2}$ connected via an into-bridge rule $i : B \xrightarrow{\sqsubseteq} i : H$
   (a) retrieve a set S_B of instances of B in ontology 1 invoking of $\mathbf{dRetrive}_1(B)$
   (b) apply to elements of S_B a transformation function f_{12} and collect transformed instances in a set S_H
   (c) $S^{dist} \leftarrow S^{dist} \cup S_H$
5. return $S^{local} \cup S^{dist}$

In order to realize the step 3 of retrieving local instances from an A-box with respect to a distributed terminology, we employ one of the approaches mentioned in Section 3, which allow to reduce reasoning with A-box to a pure terminological reasoning. This gives the possibility to seamlessly use a distributed terminological reasoner in standard Description Logics A-box reasoning algorithms.

The described algorithm can be generalized to a case of acyclical distributed knowledge bases, in which the set of indexes I is a partial order $\langle I, < \rangle$ such that $i < j$ if and only if $\mathfrak{B}_{ij} \neq \emptyset$. Dealing with cycles poses additional requirements on the algorithm such as development of cycle detection and cycle blocking strategies which we postpone to the future work.

7 Implementation

The introduced query answering algorithm is implemented on top of DRAGO distributed reasoning system [2]. DRAGO organizes a set of Reasoning Peers (DRPs) into a peer-to-peer network. Each peer is capable of providing local and distributed terminological reasoning services for ontologies with semantic mappings attached to it. Local reasoning services are implemented in accordance with standard Description Logics without considering mappings, while the distributed reasoning services theoretically grounded on the Distributed Description Logics framework as we have briefly reviewed in this paper.

We have extended Reasoning Peers with a new functionality allowing for performing instance retrieval. Similarly to the process of attaching mappings to ontology, we added a functionality of attaching instance transformation relations, which are partial specification of domain relations on the semantics level.

Figure 5 depicts a sample extract of such specification given in XML format. This simple specification, in particular, says that the distributed instance retrieval algorithm, when accessing ontology `http://source1`, should transform all instances of concept `http://source1#ConceptOne` invoking the java function `javaFunctionOne`, and all instances of concept `http://source1#ConceptTwo` invoking the java function `javaFunctionTwo`.

7.1 A motivating scenario in work

Let us step-by-step "run" an illustrative scenario that we have given at the beginning of the paper. We associate to a university human resource office a

```
<drago-dr-config>
   <domain-relations>
      <domain-relation sourceOntology="http://source1">
         <instance-transformations>
            <instance-transformation
              sourceConcept="http://source1#ConceptOne" value="${function.javaFunctionOne}"
            />
            <instance-transformation
              sourceConcept="http://source1#ConceptTwo" value="${function.javaFunctionTwo}"
            />
            ...
```

Fig. 5. Partial domain relation specification in XML format

reasoning peer DRP^{Uni}, and to a department office a DRP^{Dept}. Further, the following steps need to be performed:

- assign ontology O^{Dept} to DRP^{Dept}
- define semantic mapping $M^{Dept-Uni}$ between overlapping concepts of ontologies O^{Dept} and O^{Uni}. For example, we can have the statement that

$$O^{Dept} : PersonContact \xrightarrow{\sqsubseteq} O^{Uni} : PersonContact$$

- define individual level transformations $f^{Dept-Uni}$. For example, we can have the statement that

```
<domain-relation sourceOntology="O^{Dept}">
   <instance-transformation
      sourceConcept="O^{Dept} : PersonContact"
      value="${function.JavaEmailExtractor}"
```

  where `JavaEmailExtractor` is a function in java class that is capable of downloading the web pages, parsing the text and extracting from it e-mail addresses.
- assign ontology O^{Uni} with mapping $M^{Dept-Uni}$ and instance transformation $f^{Dept-Uni}$ to DRP^{Uni}
- notify the DRP^{Uni} that ontology O^{Dept} in $M^{Dept-Uni}$ and $f^{Dept-Uni}$ is supported by DRP^{Dept}
- start the peers O^{Uni} and O^{Dept}.

From that moment, both of the reasoning peers are ready to respond to reasoning and querying questions. When one invokes the DRP^{Uni} with a question `dRetrieve(`$PersonContact$`)` it proceeds according to the instance retrieval algorithm locally, then submits retrieval query to DRP^{Dept} as to the peer that is responsible for O^{Dept}, performs instance transformations invoking the necessary java code, and results with a set of instances.

8 Discussion and outlook

From the practical point of view, the problem of querying heterogeneous data repositories has been addressed in a variety of systems for mediation and integration of distributed heterogeneous data repositories, in particular such as Somewhere Peer Data Management System [20, 21], Piazza [22], OBSERVER [23] and others.

This article is a first attempt to understand the problem of instance retrieval from a set of heterogeneous ontologies with repositories of heterogeneous populations in the framework of Distributed Description Logics (DDLs). The previous studies of DDLs demonstrated its reason-able applicability to the problem of resolving ontological heterogeneity on a conceptual level. The presented work proposes an extension to DDLs for dealing not only with heterogeneous conceptualizations via bridge rules, but also with heterogeneous data populations via individual correspondences. Having so, allowed us to define and implement a simple distributed instance retrieval procedure for assembling instances from heterogeneous ontologies instantiated in semantically related domains.

The paper represents a work in progress. Thus it has many open issues for the future research directions. In particular, we are going to upgrade the algorithm for dealing with cyclical distributed knowledge bases. As well, we are planing to investigate the possibility of supporting more complex queries adopting the according techniques that has been developed in Description Logics community.

Acknowledgments

The work presented in the paper was partially supported by Knowledge Web FP6-507482 Network of Excellence project funded by European Union Sixth Framework Programme.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider, editors. *The Description Logic Handbook*, Cambridge University Press, 2003.
2. L. Serafini and A. Tamilin, "DRAGO: Distributed Reasoning Architecture for the Semantic Web", In *Proc. ESWC-2005*, vol 3532 of *LNCS*, pages 361–376. Springer, 2005.
3. A. Borgida and L. Serafini. "Distributed description logics: Assimilating Information from Peer Sources", *J. of Data Semantics*, 1:153–184, 2003.
4. L. Serafini, A. Borgida and A. Tamilin. "Aspects of Distributed and Modular Ontology Reasoning", In *Proc. of IJCAI-2005*, 2005.
5. I. Horrocks, U. Sattler and S. Tobies. "Practical Reasoning for Expressive Description Logics", In *Proc. LPAR-1999*, vol. 1705 of LNAI, pages 161–180. Springer, 1999.
6. I. Horrocks, U. Sattler and S. Tobies. "Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$", In *Proc. CADE-2000*, vol. 1831 of *LNCS*, pages 482–496. Springer, 2000.

7. I. Horrocks and U. Sattler. “Ontology Reasoning in the $\mathcal{SHOQ}(\mathcal{D})$ Description Logic”, In *Proc. IJCAI-2001*, pages 199–204, 2001.
8. I. Horrocks and U. Sattler. “A Tableaux Decision Procedure fro $\mathcal{SHOIQ}$”, In *Proc. IJCAI-2005*, 2005.
9. V. Haarslev and R. Möller. “Expressive ABox Reasoning with Number Restrictions, Role Hierarchies, and Transitively Closed Roles”, In *Proc. KR-2000*, 2000.
10. S. Tessaris. “Questions and Answers: Reasoning and Querying in Description Logic”, *PhD Thesis*, Department of Computer Science, University of Manchester, 2001.
11. S. Tobies. “Complexity Results and Practical Algorithms for Logics in Knowledge Representation”, *PhD Thesis*, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.
12. S. Tobies. “The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics”, *J. of Artificial Intelligence Research*, vol. 12, pages 199–217, 2000.
13. S. Tessaris and I. Horrocks. “Abox Satisfiability Reduced to Terminological Reasoning in Expressive Description Logics”, In *Proc. LPAR-2002*, 2002.
14. I. Horrocks and S. Tessaris. “Querying the Semantic Web: a Formal Approach”, In *Proc. ISWC-2002*, 2002.
15. I. Horrocks, L. Li, D. Turi and S. Bechhofer. “The Instance Store: DL Reasoning with Large Numbers of Individuals”, In *Proc. DL-2004*, 2004.
16. C. Chen, V. Haarslev and J. Wang. “LAS: Extending Racer by a Large Abox Store”, In *Proc. DL-2005*, 2005.
17. V. Haarslev and R. Möller. “RACER System Description”, In *Proc. IJCAR-2001*, vol. 2083 of *LNAI*, pages 701–705. Springer, 1998.
18. I. Horrocks and P. F. Patel-Schneider. “FaCT and DLP: Automated Reasoning with Analytic Tableaux and related Methods”, In *Proc. TABLEAUX-1998*, pages 27–30, 1998.
19. E. Sirin and B. Parsia. “Pellet: An OWL DL Reasoner”, In *Proc. ISWC-2004*, 2004. *http://www.mindswap.org/2003/pellet/index.shtml*
20. F. Goasdoué and M-C. Rousset. “Querying Distributed Data through Distributed Ontologies: a Simple but Scalable Approach”, In *J. Intelligent Systems*, vol.18 no.5, pages 60–65. IEEE, 2003.
21. M-C. Rousset. “Small can be beautiful in the Semantic Web”, In *Proc. ISWC-2004*, vol.3298 of *LNCS*, pages 6–16. Springer, 2004.
22. A. Halevy, Z. Ives, P. Mork and I. Tatarinov. “Piazza: Data Management Infrastructure for Semantic Web Applications”, In *Proc. of WWW-2003*, 2003.
23. E. Mena, A. Illarramendi, V. Kashyap and A. Sheth. ”OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies”, In *J. on Distributed and Parallel Databases*, vol. 8 n. 2, pages 223–272, 2000.
24. P. Pothipruk and G. Governatori. ”A Formal Ontology Reasoning with Individual Optimization: A Realization of the Semantic Web”, In *Proc. WISE-2005*, pages 119–132, 2005.