Reasoning about Ontology Mappings

Heiner Stuckenschmidt[1], Luciano Serafini [2] and Holger Wache[3]

Abstract. The use of logic-based representations in distributed environments such as the semantic web has led to work on the representation of and reasoning with mappings between distributed ontologies. Up to now the investigation of reasoning methods in this area was restricted to the use of mapping for query answering or subsumption reasoning. In this paper, we investigate the task of reasoning about the mappings themselves. We identify a number of properties such as consistency and entailment of mappings that are important for validating and comparing mappings. We provide formal definitions for these properties and show how the properties can be checked using existing reasoning methods by reducing them to local and global satisfiability testing in distributed description logics.

Keywords: Ontologies, Mappings, Reasoning, Consistency

1 MOTIVATION

The problem of semantic heterogeneity is becoming more and more pressing in many areas of information technologies. The Semantic Web is only one area where the problem of semantic heterogeneity has lead to intensive research on methods for semantic integration. The specific problem of semantic integration on the Semantic Web is the need to not only integrate data and schema information, but to also provide means to integrate ontologies, rich semantic models of a particular domain. There are two lines of work connected to the problem of a semantic integration of ontologies:

- The (semi-) automatic detection of semantic relations between ontologies (e.g. [10, 7, 13, 14, 8]).
- The representation and use of semantic relations for reasoning and query answering (e.g. [15, 12, 6, 4, 5]).

So far, work on representation of and reasoning with mappings has focussed on mechanisms for answering queries and using mappings to compute subsumption relationships between concepts in the mapped ontologies. These methods always assumed that the mappings used are manually created and of high quality (in particular consistent). In this paper we investigate logical reasoning about mappings that are not assumed to be perfect. In particular, our methods can be used to check (automatically created) mappings for formal and conceptual consistency and determine implied mappings that have not explicitly been represented. We investigate such mappings in the context of distributed description logics [3, 17], an extension of traditional description logics with mappings between concepts in different T-Boxes. The functionality described in this paper will become more important in the future because more and more ontologies are created and need to be linked. For larger ontologies the process of mapping will not be done completely by hand, but will rely on or will at least be supported by automatic mapping approaches. We see our work as a contribution to semi-automatic approaches for creating mappings between ontologies where possible mappings are computed automatically and then corrected manually making use of methods for checking the formal and conceptual properties of the mappings. The concrete contributions of this paper are the following:

- We define a number of formal properties that mappings should satisfy
- We present methods for checking these properties by rephrasing them as reasoning problems in distributed description logics
- We discuss how the properties can be tested using basic reasoning methods for distributed description logics implemented in the DRAGO reasoning system[16].

The paper is organized as follows. In section 2 we briefly review distributed description logics (DDL) as an extension of traditional description logics and discuss reasoning in this logic. In section 3 we introduce a number of formal properties that mappings in DDL should satisfy. Methods for checking these properties by rephrasing it to a reasoning problem in DDL are presented in section 4 and conclude with a discussion in section 5.

2 RELATED WORK

Several researchers have proposed frameworks for describing mappings on an abstract level. These frameworks try to capture general aspects of mappings, often independent of a particular encoding language or an intended use. Calvanese and others [6] describe a framework for mapping ontologies that is motivated by their previous work on database integration. The framework is based on the traditional database integration architecture with a global and several local models and re-applies common database notions like the Global-as-View and and Local-As-View approach to integration. The work of Madhavan and others [12] is also inspired by the database integration problem but allows more flexible architectures than the Calvanese paper. The general framework consists of some core definitions and a number of reasoning problems that are illustrated in the relational framework. In the context of the European Network KwowledgeWeb a general framework for the representation of mappings between semantic models has been developed [5]. The framework in intentionally independent of a particular representation language and only defines different types and elements of mappings.

[1] University of Mannheim, Germany, email: heiner@informatik.uni-mannheim.de

[2] ITC-IRST, Trento, Italy, email: serafini@itc.it

[3] Vrije Universiteit Amsterdam, Netherlands, email: holger@cs.vu.nl

In the context of description logics which are the most common formalism for specifying ontologies, there are currently two formalisms that have been proposed for representing semantic relations between ontologies that are in the ficus of interest. The approach presented in [3] extends DL with a local model semantics similar to the one introduced above and so-called bridge rules to define semantic relations between different T-Boxes. A distributed interpretation for DDL on a family of DL language $\{L_i\}$, is a family $\{\mathcal{I}_i\}$ of interpretations, one for each L_i plus a family $\{r_{ij}\}_{i \neq j \in I}$ of domain relations. While the original proposal only considered subsumption between concept expressions, the model was extended to a set of five semantic relations: Equivalence, Disjointness, Overlap and Containment in two directions. A similar approach for defining relations between DL knowledge bases has emerged from the investigation of so-called ϵ-connections between abstract description systems [11]. Originally intended to extend the decidability of DL models by partitioning it into a set of models that use a weaker logic, the approach has recently been proposed as a framework for defining mappings between ontologies [9]. In the ϵ-connections framework, for every pair of ontologies ij there is a set ϵ_{ij} of *links*, which represents binary relations between the domain of the i-th ontology and the domain of the j-th ontology. Links from i to j can be used to define i concepts, in a way that is analogous to how roles are used to define concepts. In the following table we report the syntax and the semantics of i-concepts definition based on links. (E denotes a link from i to j and C denotes a concept in j. It has been shown that distributed description logics can be seen as a special case of ϵconnections where links between two ontologies are interpreted as a weak form of subsumption.

In our work, we take distributed description logics as the basis for discussing mappings. Due to the tight relationship between distributed description logics and ϵ-connections some of the definitions will also apply in this general model.

3 DISTRIBUTED DESCRIPTION LOGICS

Distributed Description Logics as proposed in [3] provide a language for representing sets of terminologies. For this purpose DDLs provide mechanisms for referring to terminologies and for defining rules that connect concepts in different terminologies. On the semantic level, DDLs extend the standard notion of interpretation for description logics (compare [1]) to fit the distributed nature of the model and to reason about concept subsumption across terminologies.

Let I be a non-empty set of indices and $\{\mathcal{T}_i\}_{i \in I}$ a set of terminologies. We prefix inclusion axioms with the index of the terminology they belong to (i.e. $i : C$ denotes a concept in terminology $\mathcal{T}_i$ and $j : C \sqsubseteq D$ a concept inclusion axiom from terminology $\mathcal{T}_j$). Note that $i : C$ and $j : C$ are different concepts. Semantic relations between concepts in different terminologies are represented in terms of axioms of the following form, where C and D are concepts in terminologies $\mathcal{T}_i$ and $\mathcal{T}_j$, respectively:

- $i : C \xrightarrow{\sqsubseteq} j : D$ (into)
- $i : C \xrightarrow{\sqsupseteq} j : D$ (onto)
- $i : C \xrightarrow{\equiv} j : D$ (equivalence)
- $i : C \xrightarrow{\perp} j : D$ (disjointness)

These axioms are called *bridge-rules*. A distributed terminology $\mathfrak{T}$ is now defined as a pair $(\{\mathcal{T}_i\}_{i \in I}, \{\mathcal{B}_{ij}\}_{i \neq j \in I})$ where $\{\mathcal{T}_i\}_{i \in I}$ is a set of terminologies and $\{\mathcal{B}_{ij}\}_{i \neq j \in I}$ is a set of bridge rules between these terminologies.

The semantics of distributed description logics is defined in terms of a distributed interpretation $\mathfrak{I} = (\{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I})$ where $\mathcal{I}_i = (\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i})$ is an interpretation for T-Box $\mathcal{T}_i$ as used in Description Logics or an interpretation on the empty domain that maps each concept and role on the empty set (compare [17]) and $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ is a domain relation connecting elements of the interpretation domains of terminologies $\mathcal{T}_i$ and $\mathcal{T}_j$. We use $r_{ij}(x)$ to denote $\{y \in \Delta^{\mathcal{I}_j} | (x, y) \in r_{ij}\}$ and $r_{ij}(C)$ to denote $\bigcup_{x \in C} r_{ij}(x)$.

A distributed interpretation $\mathfrak{I}$ satisfies a distributed terminology $\mathfrak{T}$ if:

- $\mathcal{I}_i$ satisfies $\mathcal{T}_i$ for all $i \in I$
- $r_{ij}(C^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_j}$ for all $i : C \xrightarrow{\sqsubseteq} j : D$ in $\mathcal{B}_{ij}$
- $r_{ij}(C^{\mathcal{I}_i}) \supseteq D^{\mathcal{I}_j}$ for all $i : C \xrightarrow{\sqsupseteq} j : D$ in $\mathcal{B}_{ij}$
- $r_{ij}(C^{\mathcal{I}_i}) = D^{\mathcal{I}_j}$ for all $i : C \xrightarrow{\equiv} j : D$ in $\mathcal{B}_{ij}$
- $r_{ij}(C^{\mathcal{I}_i}) \cap D^{\mathcal{I}_j} = \emptyset$ for all $i : C \xrightarrow{\perp} j : D$ in $\mathcal{B}_{ij}$

In this case we call $\mathfrak{I}$ a model for $\mathfrak{T}$. A concept $i : D$ subsumes a concept $i : C$ ($i : \{C \sqsubseteq D\}$) if for all models of $\mathfrak{T}$ we have $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$. A concept $i : C$ is inconsistent if $\mathfrak{T} \models i : \{C \equiv \perp\}$

Reasoning in DDL differs from reasoning in traditional description logics by the way knowledge is propagated between T-Boxes by certain combinations of bridge rules. The simplest case in which knowledge is propagated is the following:

$$\frac{i : A \xrightarrow{\sqsupseteq} j : G, \quad i : B \xrightarrow{\sqsubseteq} j : H, \quad i : \{A \sqsubseteq B\}}{j : \{G \sqsubseteq H\}} \tag{1}$$

This means that the subsumption between two concepts in a terminology can depend on the subsumption between two concepts in a different terminology if the subsumed concepts are linked by the onto- and the subsuming concepts by an into-rule. In languages that support disjunction, this basic propagation rule can be generalized to subsumption between a concept and a disjunction of other concepts in the following way:

$$\frac{\begin{array}{c} i : A \xrightarrow{\sqsupseteq} j : G, i : \{A \sqsubseteq B_1 \sqcup \cdots \sqcup B_n\} \\ i : B_1 \xrightarrow{\sqsubseteq} j : H_1, \ldots, i : B_n \xrightarrow{\sqsubseteq} j : H_n, \end{array}}{j : \{G \sqsubseteq H_1 \sqcup \cdots \sqcup H_n\}} \tag{2}$$

It has been shown that this general propagation rule completely describes reasoning in DDL that goes beyond well known methods for reasoning in Description Logics [17]. To be more specific, adding the inference rule in equation 2 to existing tableaux reasoning methods leads to a correct and complete method for deciding subsumption in DDL. The method has been implemented in the DRAGO system [16] which is available for download at `http://drago.itc.it/`.

4 PROPERTIES OF MAPPINGS

The formal semantics of distributed description logics tells us how to reason about concepts in a distributed T-Box taking into account the constraints on the interpretation imposed by mappings (sets of bridge rules) by means of formal properties like subsumption and inconsistency. These properties have been proven useful to support the development of high quality centralized ontologies [2]. When extending centralized to distributed ontologies by means of mappings, there is a need for similar concepts to support the development of high quality mappings. In this context, we have to define properties that reflect the quality of a mapping and can be tested by formal reasoning. In this section, we introduce four properties that reflect the quality of a mapping, namely *containment*, *minimality*, *consistency* and *embedding*. In the following, we explain these properties and their connection to mapping quality and provide a formal characterization of each of the properties that will be used to define effective methods for checking these properties using the DRAGO reasoning system.

4.1 Consistency and Embedding

The first two properties we will discuss can be seen as the counterpart of the notion of satisfiability of a concept or a T-Box for mappings. In particular, we want to test whether a set of bridge rules make sense from a conceptual point of view. We start with an example. Let $\mathfrak{T}$ be a distributed T-Box composed of the two terminologies $\mathcal{T}_i$ and $\mathcal{T}_j$ with the mappings $\mathcal{B}_{ij}$ as displayed in figure 1.

It can easily be shown that by applying the definition of satisfiability of bridge rules, any distributed interpretation for $\mathfrak{T}$ is such that $\mathsf{BaStudent}^{\mathcal{I}_j} = \emptyset$. Clearly this is not a desirable property for a mapping. It means that the additional constraints on the interpretation induced by the bridge rules are too strong as they make parts of the target terminology unsatisfiable. In this case the mappings can be fixed by weakening the first bridge rule to $i:\mathsf{Student} \xrightarrow{\sqsubseteq} j:\mathsf{Student}$.

In order to avoid situations like the one above, we introduce the notion of consistency for mappings and claim that a mapping is consistent if it does not make a satisfiable concept in the target terminology unsatisfiable:

Definition 1 (Consistency) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i\in I}, \{\mathcal{B}_{ij}\}_{i\neq j\in I})$ be a distributed terminology. The mappings $\{\mathcal{B}_{ij}\}_{i\neq j\in I}$ of $\mathfrak{T}$ are consistent if for all atomic concepts C in any of the terminologies $\mathcal{T}_i$ such that $\mathcal{T}_i \not\models C \equiv \bot$ we have $\mathfrak{T} \not\models C \equiv \bot$. The mappings of a distributed terminology are called inconsistent if they are not consistent.*

The notion of consistency is useful for evaluating mappings that have been generated automatically using ontology matching tools. As most of the existing tools are based on heuristics and do not check logical implications of mappings, a situation like the above can occur with generated mappings. Checking consistency in the sense of the definition above will detect unwanted effects of these mappings.

There are cases where combinations of bridge rules have an unwanted effect even though they do not fall under the notion of inconsistency introduced above, because they do not make any concept unsatisfiable. Consider the following pair of bridge rules:

$$i:\mathsf{Car} \xrightarrow{\sqsubseteq} j:\mathsf{UsefulThing} \tag{3}$$

$$i:\mathsf{Car} \xrightarrow{\sqsubseteq} j:\mathsf{UselessThing} \tag{4}$$

If in the j-th terminology useful and useless things are defined as disjoint classes as we would expect ($\mathsf{UselessThing} \sqsubseteq \neg\mathsf{UsefulThing}$), then our intuition is that these two mappings cannot jointly be satisfiable. According to definition 1, however, they are consistent. The reason is that the semantics of DDL admits the situation in which r_{ij} is not defined on any element of $\mathsf{Car}^{\mathcal{I}_i}$. In this case, a model for the situation above can be constructed such that: $r_{ij}(\mathsf{Car}^{\mathcal{I}_i}) = \emptyset \subseteq \mathsf{UsefulThing}^{\mathcal{I}_j}$ and $r_{ij}(\mathsf{Car}^{\mathcal{I}_i}) = \emptyset \subseteq \mathsf{UselessThing}^{\mathcal{I}_j}$. So there exists at least one satisfiable interpretation $\mathfrak{I}$.

However a satisfiable mapping on the empty set is not desirable for practical reasons. Mappings are useful when they can be used to transfer information from one terminology to the other. For instance, the mapping $i : A \xrightarrow{\sqsubseteq} j : B$ transfers the fact of x being A in terminology $\mathcal{T}_i$, into the fact that $r_{ij}(x)$ is B in $\mathcal{T}_j$. If the domain relation r_{ij} is empty, then no information is transferred, and therefore, despite the consistency of mappings, they are still useless.

To catch the above intuition of a mapping that has the capability of transferring data from i to j we define the notion of an embedding in the following way.

Definition 2 (Embedding) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i\in I}, \{\mathcal{B}_{ij}\}_{i\neq j\in I})$ be a distributed terminology. The mappings $\{\mathcal{B}_{ij}\}_{i\neq j\in I}$ are embedding mappings if for all atomic concepts C in any of the terminologies $\mathcal{T}_i$ with $\mathcal{T}_i \not\models C \equiv \bot$ there is a distributed interpretation $\mathfrak{I}$ such that for all $\mathcal{T}_j$ we have $r_{ij}(C_i^{\mathcal{I}}) \neq \emptyset$.*

According to the definition we have a set of bridge rules that contains (3) and (4) is not an embedding as it can be satisfied only if $r_{ij}(\mathsf{Car}) = \emptyset$. This property is another one that can be used to check the output of automatic mapping approaches on a logical level in order to ensure that the resulting mapping can actually be used to transfer information between the terminologies that have been mapped.

There are some problems with this global definition of embedding of a complete mapping as there are cases where the inability to transfer information between models is not actually a bug, but represents the different viewpoints taken by each model. Consider the case where $\mathcal{T}_j$ is a terminology that speaks about food and contains the axioms

$$\mathsf{toxic} \equiv \neg\mathsf{eatable} \tag{5}$$

i.e., they only want to distinguish between toxic and eatable things. If terminology $\mathcal{T}_i$ speaks about the things sold in a big super-store, which sells food, computers and plants, then it will contain object which are neither toxic nor eatable, say for instance flowers, and the following mappings would be acceptable,

$$i:\mathsf{FreshMilk} \xrightarrow{\sqsubseteq} j:\mathsf{Eatable} \tag{6}$$

$$i:\mathsf{OldMilk} \xrightarrow{\sqsubseteq} j:\mathsf{Toxic} \tag{7}$$

$$i:\mathsf{Rose} \xrightarrow{\perp} j:\mathsf{Eatable} \tag{8}$$

$$i:\mathsf{Rose} \xrightarrow{\perp} j:\mathsf{Toxic} \tag{9}$$

Clearly mapping (8) and (9) together with the axiom (5), entails that $r_{ij}(\mathsf{Rose}^{\mathcal{I}_i}) = \emptyset$. But in this example this fact is acceptable, since the second terminology is supposed not to have anything that corresponds to a rose.

To accommodate with this case we can refine the definition of embedding to refer to a certain concept on which we require the domain relation to be defined.

Axioms of $\mathcal{T}_i$	Axioms of $\mathcal{T}_j$
$i:\mathsf{Student} \equiv \mathsf{PhdStudent} \sqcup \mathsf{MsStudent}$ $i:\mathsf{PhdStudent} \sqsubseteq \neg\mathsf{MsStudent}$	$j:\mathsf{Student} \equiv \mathsf{PhdStudent} \sqcup \mathsf{MsStudent} \sqcup \mathsf{BaStudent}$ $j:\mathsf{PhdStudent} \sqsubseteq \neg\mathsf{MsStudent}$ $j:\mathsf{BaStudent} \sqsubseteq \neg\mathsf{MsStudent}$ $j:\mathsf{PhdStudent} \sqsubseteq \neg\mathsf{BaStudent}$

Mappings from $\mathcal{T}_i$ to $\mathcal{T}_j$ in $\mathcal{B}_{ij}$
$i:\mathsf{Student} \xrightarrow{\equiv} j:\mathsf{Student}$ $i:\mathsf{PhDStudent} \xrightarrow{\equiv} j:\mathsf{PhDStudent}$ $i:\mathsf{MsStudent} \xrightarrow{\equiv} j:\mathsf{MsStudent}$

Figure 1. Mapping Example

Definition 3 (Embedding for a concept) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i\in I}, \{\mathcal{B}_{ij}\}_{i\neq j\in I})$ be a distributed terminology. The mappings $\{\mathcal{B}_{ij}\}_{i\neq j\in I}$ are an embedding for an atomic concept C in terminology $\mathcal{T}_i$ if $\mathcal{T}_i \not\models C \equiv \bot$ implies that there is a distributed interpretation $\mathfrak{I}$ such that for all $\mathcal{T}_j$ we have $r_{ij}(C_i^{\mathcal{I}}) \neq \emptyset$. The mappings are an embedding in the sense of definition 2 if they are an embedding for all concepts in $\mathfrak{T}$.*

This refined version of embedding provides us with a powerful analytical tool that ontology engineers can use to assess the quality of mappings and also to better understand differences in the viewpoints taken by different terminologies. Computing the set of non-embedded concepts gives us an idea of topics on which two terminologies take different points of view. On the other hand we can state expectations about differences in viewpoints by specifying sets of concepts that we assume to be embedded or non-embedded respectively. Based on this assumption, we can test whether the mapping actually reflects this assumption.

4.2 Containment and Minimality

The remaining two properties to be discussed here can be seen as the counterpart of subsumption in classical Description Logics applied to mappings. In particular, these properties are closely connected to the notion of entailment between bridge rules. Consider the following two rules.

$$i:\mathsf{Car} \xrightarrow{\sqsubseteq} j:\mathsf{Vehicle} \tag{10}$$

$$i:\mathsf{SportCar} \xrightarrow{\sqsubseteq} j:\mathsf{Vehicle} \tag{11}$$

Supposed that $\mathcal{T}_i$ contains the axiom $\mathsf{SportCar} \sqsubseteq \mathsf{Car}$. Mapping (11) is redundant, as it is already contained in the mapping (10). In other words, mapping (11) is entailed by mapping (10) and the axioms of $\mathcal{T}_i$. In the following definition we formalize the notion of entailment (or consequence) between mappings

Definition 4 (Entailment) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i\in I}, \{\mathcal{B}_{ij}\}_{i\neq j\in I})$ be a distributed terminology. A bridge rule $i:A \xrightarrow{R} j:B$ ($i \neq j, R \in \{\sqsubseteq, \sqsupseteq, \bot, \equiv\}$) is entailed by $\mathfrak{T}$ if every model $\mathfrak{I} = (\{\mathcal{I}_i\}_{i\in I}, \{r_{ij}\}_{i\neq j\in I})$ of $\mathfrak{T}$ satisfies $i:A \xrightarrow{R} j:B$ (compare sec. 3).*

Entailment of bridge rules can be used to compute and evaluate the consequences of a mapping. Existing mapping approaches normally use heuristics to prune the search space for possible mappings and therefore do not test each combination of concepts for a possible semantic correspondence. In the case of the example above, most mapping approaches would only compute $i:\mathsf{Car} \xrightarrow{\sqsubseteq} j:\mathsf{Vehicle}$. The notion of entailment allows us to check that this also covers the mapping from sports cars to vehicles as we would expect.

Another application of this property is to compute direct mappings between terminologies that are only connected via paths of mappings in the explicit model. This corresponds to the composition of existing mappings and increases the effectiveness of distributed reasoning by creating mapping shortcuts that can directly be used in following reasoning steps. In the above example $\mathcal{T}_j$ contains knowledge about House. Obviously House and $\mathsf{Vehicle}$ are declared to be disjoint, i.e. $\mathsf{House} \sqsubseteq \neg\mathsf{Vehicle}$. Suppose a third terminology $\mathcal{T}_k$ joins with the following rule

$$j:\mathsf{House} \xrightarrow{\sqsupseteq} k:\mathsf{Flat} \tag{12}$$

then the shortcut $i:\mathsf{Car} \xrightarrow{\bot} k:\mathsf{Flat}$ is entailed by $\mathfrak{T}$. In general entailment allows to conclude not only between several terminologies but can also use knowledge in the local terminologies.

Based on this notion of entailment, we can introduce two additional properties of mappings that are useful in the context of evaluating ontology mappings. Containment says that one mapping logically follows from another one, Minimality refers to the most compact representation of a mapping.

Definition 5 (Containment and Minimality) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i\in I}, \{\mathcal{B}_{ij}\}_{i\neq j\in I})$ be a distributed terminology. A set of mappings $\{\mathcal{B}'_{ij}\}_{i\neq j\in I}$ is* contained *in $\{\mathcal{B}_{ij}\}_{i\neq j\in I}$ if and only if for each bridge rule $b \in \{\mathcal{B}'_{ij}\}_{i\neq j\in I}$ b is entailed by $\mathfrak{T}$. A set of bridge rules $\{\mathcal{B}'_{ij}\}_{i\neq j\in I}$ is* minimal*, if there is no subset $\{\mathcal{B}''_{ij}\}_{i\neq j\in I}$ of $\mathcal{B}'_{ij}$ such that $\{\mathcal{B}'_{ij}\}_{i\neq j\in I}$ is contained in $\{\mathcal{B}''_{ij}\}_{i\neq j\in I}$.*

The notion of minimality is important when it comes to comparing the results of automatic mapping systems in terms of precision and recall. Such an evaluation is normally done by comparing the results of different systems to a gold standard mapping. In order to guarantee a fair evaluation, only the minimal representations of all mappings should be compared because otherwise approaches that compute more mappings than necessary will get a penalty in terms of precision.

5 DECIDING MAPPING PROPERTIES

In order to use the criteria defined above for engineering and evaluating mappings between terminological models, we need efficient methods for deciding whether these properties hold in a given setting. In this section, we show that all of the properties can be tested using existing reasoning methods for distributed description logics. Given a distributed T-Box, $\mathfrak{T}$, the following reasoning services are available in the DRAGO system:

- *Local/global satisfiability:* check if $\mathcal{T}_i \models C \equiv \bot$, and $\mathfrak{T} \models i : C \equiv \bot$
- *Local/global subsumption:* check if $\mathcal{T}_i \models C \sqsubseteq D$, and $\mathfrak{T} \models i : C \sqsubseteq D$
- *Local/global classification:* Produce a classification on the atomic concepts of O_i. A classification on a set of atomic concepts $\mathcal{C}$, is directed acyclic graph $\langle \mathcal{C}, \prec, \sim \rangle$, where $\mathcal{C}$ is the set of atomic concepts of the language of $\mathcal{T}_i$ and $\prec$ constitute a directed acyclic graph on $\mathcal{C}$, and $\sim$ is an equivalence relation on $\mathcal{C}$. And the following properties holds, $C \sim D$ iff $\mathcal{T}_i \models C \equiv D$ (resp $\mathfrak{T} \models i : C \equiv D$), $C \prec D$ if and only if $\mathcal{T}_i \models C \sqsubseteq D$ and $\mathcal{T}_i \not\models C \sqsubseteq D$, (resp. $\mathfrak{T} \models i : C \sqsubseteq D$ and $\mathfrak{T} \not\models i : C \sqsubseteq D$). Furthermore if $C \prec D$ then for no $E \in \mathcal{C}$, $C \prec E \prec D$

In the following we show a simple (not optimized, but viable) way to check the properties introduced in the previous section, by using these reasoning services.

5.1 Consistency

A procedure for checking consistency of a mapping can be obtained by a direct application of the definition: In particular by checking whether all locally satisfiable concepts are also globally satisfiable.

```
CONSISTENCYCHECK($\mathfrak{T} = (\{\mathcal{T}_i\}_{i \in I}, \{\mathcal{B}_{ij}\}_{i \neq j \in I})$) computes if
the mappings $\{\mathcal{B}_{ij}\}_{i \neq j \in I}$ are consistent w.r.t. the distributed terminology $\mathfrak{T}$
1. GLOBALCLASSIFY$_j(\mathfrak{T})$
2. if for some $C$ in any of the terminologies $\mathcal{T}_i$, such that $C \equiv \bot$,
   LOCALSAT$_i(C)$ = TRUE then return FALSE else return TRUE
```

The soundness and completeness of CONSISTENCYCHECK is guaranteed by the soundness and completeness of the function GLOBALCLASSIFY$_j$.

5.2 Embedding

The notion of embedding of a concept cannot directly be checked using the available reasoning services because the definition of embedding does not rely on the satisfiability of a concept, but on the image $r_{ij}(C)$. In order to be able to use the reasoning services, we have to make this image explicit by turning it into a new named concept in the target terminology.

Definition 6 (Image) *The j-image of a concept C from terminology $\mathcal{T}_i$ in $\mathcal{T}_j$ is a concept $C_j^{\rightarrow}$ not already in $\mathcal{T}_j$ that is defined by:*

1. $j : C_j^{\rightarrow} \sqsubseteq \top$
2. $i : C \xrightarrow{\equiv} j : C_j^{\rightarrow}$

We denote the terminology $\mathcal{T}_j$ extended with (1) as $\mathcal{T}_j^{C^{\rightarrow}}$ and the set of mappings extended with (2) as $\mathcal{B}_{ij}^{C^{\rightarrow}}$. We further denote the distributed terminology resulting from extending it with all possible j-images of a concept C as $\mathfrak{T}^{C^{\rightarrow}}$ and the distributed terminology extended by the definitions of all possible images of the concepts in all terminologies as $\mathfrak{T}^{\rightarrow}$

This notion of an image allows us to directly ask questions about the semantic relation of the image of a concept to other concepts in the target terminology. This means that we can reformulate the embedding property in terms of conditions that only apply to named concepts in the following way.

Definition 7 (Embedding) *Let $\mathfrak{T} = (\{\mathcal{T}_i\}_{i \in I}, \{\mathcal{B}_{ij}\}_{i \neq j \in I})$ be a distributed terminology. The mappings $\{\mathcal{B}_{ij}\}_{i \neq j \in I}$ are an embedding for a concept C in any of the terminologies $\mathcal{T}_i$ with $\mathcal{T}_i \not\models C \equiv \bot$ if $\mathfrak{T}^{C^{\rightarrow}} \not\models C_j^{\rightarrow} \equiv \bot$*

A test for this notion of embedding for a concept can now be implemented using the available reasoning services for distributed description logics. A corresponding algorithm is given below.

```
EMBEDDINGCHECK($\mathfrak{T} = (\{\mathcal{T}_i\}_{i \in I}, \{\mathcal{B}_{ij}\}_{i \neq j \in I}), C$) checks if
$\{\mathcal{B}_{ij}\}_{i \neq j \in I}$ is an embedding for a concept C.
1. if LOCALSAT$_i(C)$ = TRUE and
2. GLOBALSAT($\mathfrak{T}^{C^{\rightarrow}}, j : C^{\rightarrow}$) = TRUE then return TRUE else return FALSE
```

This test for the embedding of a concept can of course easily be extended to testing the general embedding property for a mapping. In this case, we just iterate the embedding test over all concepts in the source terminology.

5.3 Entailment

The idea of explicitly representing images of concepts in the target terminology can also be used to give an operational definition for testing entailment of bridge rules. This is done by extending the distributed terminology with images of all concepts in the other terminologies and checking the semantic relation between these images and other concepts in the target terminology in the following way:

Proposition 1 (entailment) *Let $\mathfrak{T}$ be a distributed terminology. Then the following equivalences hold for any concept C in any of the terminologies $\mathcal{T}_i$*

$$
\begin{array}{lcl}
\mathfrak{T}^{\rightarrow} \models j : C_j^{\rightarrow} \equiv D & \iff & \mathfrak{T} \models i : C \xrightarrow{\equiv} j : D \\
\mathfrak{T}^{\rightarrow} \models j : C_j^{\rightarrow} \sqsubseteq D & \iff & \mathfrak{T} \models i : C \xrightarrow{\sqsubseteq} j : D \\
\mathfrak{T}^{\rightarrow} \models j : C_j^{\rightarrow} \sqsupseteq D & \iff & \mathfrak{T} \models i : C \xrightarrow{\sqsupseteq} j : D \\
\mathfrak{T}^{\rightarrow} \models j : C_j^{\rightarrow} \sqcap D \sqsubseteq \bot & \iff & \mathfrak{T} \models i : C \xrightarrow{\bot} j : D
\end{array}
$$

On the basis of the above propositions we can define the following procedure for checking consequence using the available reasoning service of the DRAGO system.

```
DERIVABILITYCHECK($\mathfrak{T}, C : i \xrightarrow{R} D : j$) verifies if the mapping $i : C \xrightarrow{R} j : D$) is a consequence
of mapping $\mathcal{B}_{ij}$ w.r.t. to a distributed terminology $\mathfrak{T}$

$R$ = " $\sqsubseteq$ " return GLOBALSUBSUMPTION($\mathfrak{T}^{\rightarrow}, j : C^{\rightarrow} \sqsubseteq D$)
$R$ = " $\sqsupseteq$ " return GLOBALSUBSUMPTION($\mathfrak{T}^{\rightarrow}, j : D \sqsubseteq C^{\rightarrow}$)
$R$ = " $\equiv$ " return GLOBALSUBSUMPTION($\mathfrak{T}^{\rightarrow}, j : D \sqsubseteq C^{\rightarrow}$) $\wedge$
    GLOBALSUBSUMPTION($\mathfrak{T}^{\rightarrow}, j : C^{\rightarrow} \sqsubseteq D$)
$R$ = "$\bot$" return $\neg$GLOBALSATISFIABLE($\mathfrak{T}^{\rightarrow}, j : D \sqcap C^{\rightarrow}$)
```

Once we can check entailment of a mapping, we can use this method to check the properties containment and minimality that are both defined based on the notion of entailment. In the case of containment, we test entailment for all mappings of the contained mapping. Minimality is somehow more complicated to check as it might require checking all possible subsets of a mapping for the containment property, but nevertheless it can be done using the reasoning services mentioned above.

6 DISCUSSION

In this paper, we discussed the problem of reasoning about formal properties of mappings between concept expressions in distributed description logics. The investigation was motivated by the need to verify and compare ontology mappings that have been created automatically or by human experts. In order to support this task, we proposed a number of formal properties of sets of mapping statements that provide an insight in the quality of a mapping. The definition of these properties was inspired by the practical need of evaluating mappings as well as by well-established formal properties of logical models such as consistency and minimality. In this the proposed properties differ from the ones proposed in [12] which primarily focus on the problem of using mappings for answering queries across different models. On an abstract level, we can say that our properties are useful at design time while the properties proposed by [12] are useful at run time.

In the paper, we also provided some preliminary results on the problem of automatically checking mapping properties. As a starting point, we used the definitions of the properties and tried to map them on methods for reasoning in DDL. The result are theoretical algorithms for deciding mapping properties based on the definition of a distributed T-Box. During the concrete implementation of the methods as a extension to the DRAGO reasoning system which is currently under way, it turned out, that many of the definitions cannot directly be transferred to the implementation. The reason for this is that the definition of a distributed T-Box as a set of T-Boxes and mappings abstracts from the actual situation by assuming that all the information is globally available. The DRAGO system, however is implemented in terms of a P2P-Architecture where each peer only knows its own model and has limited access to models in the neighborhood via mappings. This limited availability of information requires a more realistic formalization of a distributed T-Box as a basis for designing algorithms that can actually be implemented in the system.

As mentioned in the related work section, DDLs can be seen as a special case of ϵ-connections. As there are reasoners that support reasoning with ϵ-connections, another question is whether we can also use the corresponding reasoner as a basis for checking the properties proposed. This still needs to be checked along with the question if the properties proposed also make sense in the general setting of ϵ-connections, where links between ontologies do not only represent semantic relations, but can also consist of domain relations.

A general question is concerned with the complexity of algorithms for testing mapping properties. Here, we have to distinguish between the theoretical complexity of the reasoning task and the concrete complexity of different algorithms. So far we do not have exact complexity results but for the theoretical complexity it is likely that the complexity of checking subsumption in the local models dominates the complexity of the complete method. For the concrete logic implemented in the DRAGO system checking subsumption is EXP-time complete [18]. The more interesting question in this context is about the complexity of concrete algorithms that try to minimize the communication costs in the distributed implementation of the DRAGO system.

REFERENCES

[1] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook - Theory, Implementation and Applications*. Cambridge University Press, 2003.

[2] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. Oiled: a reason-able ontology editor for the semantic web. In *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*, volume Vol. 2174 of *LNAI*, pages 396–408, Vienna, Austria, September 19-21 2001. Springer-Verlag.

[3] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.

[4] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-OWL: Contextualizing ontologies. In *Second International Semantic Web Conference ISWC'03*, volume 2870 of *LNCS*, pages 164–179. Springer, 2003.

[5] Paolo Bouquet, Jerome Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Deliverable 2.2.4, KnowledgeWeb, 2004.

[6] D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *Proceedings of the Semantic Web Working Symposium*, pages 303–316, Stanford, CA, 2001.

[7] Marc Ehrig and Steffen Staab. QOM - Quick Ontology Mapping. In *3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, 2004.

[8] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Semantic matching. In *1st European semantic web symposium (ESWS'04)*, pages 61–75, Heraklion, Greece, 2004.

[9] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Working with multiple ontologies on the semantic web. In *Proceedings of the Third Internatonal Semantic Web Conference (ISWC2004)*, volume 3298 of *Lecture Notes in Computer Science*, 2004.

[10] E. Hovy. Combining and standardizing largescale, practical ontologies for machine translation and other uses. In *The First International Conference on Language Resources and Evaluation (LREC)*, pages 535–542, Granada, Spain, 1998.

[11] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[12] Jayant Madhavan, Philip A. Bernstein, Pedro Domingos, and Alon Halevy. Representing and reasoning about mappings between domain models. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI'2002)*, Edmonton, Canada, 2002.

[13] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *18th International Conference on Data Engineering (ICDE-2002)*, San Jose, California, 2002. IEEE Computing Society.

[14] N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[15] L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping languages for terminological knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence - IJCAI05*, Edingurgh, UK, August 2005.

[16] L. Serafini and A. Tamilin. DRAGO: Distributed reasoning architecture for the semantic web. In *In Proceedings of the Second European Semantic Web Conference (ESWC'05)*. Springer-Verlag, 2005.

[17] Luciano Serafini, Alex Borgida, and Andrei Tamilin. Aspects of distributed and modular ontology reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI-05*, Edinburgh, Scotland, 2005.

[18] Heiner Stuckenschmidt and Michel Klein. Reasoning and change management in modular ontologies. Technical Report TR-2005 -012, University Of Mannheim, 2005.