Reasoning about Ontology Mappings

Heiner Stuckenschmidt[1], Luciano Serafini[2], Holger Wache[3]
[1]Universitaet Mannheim, Germany
[2]ITC-IRST, Trento, Italy
[3]Vrije Universiteit Amsterdam, Netherlands

November 3, 2005

Abstract

The use of logic-based representations in distributed environments such as the semantic web has lead 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 [8, 6, 10, 11, 7].
- The representation and use of semantic relations for reasoning and query answering [5, 9, 4, 14, 3].

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 [2, 13], an extension of traditional description logics with mappings between concepts in different T-Boxes. The functionality described in this paper will be 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 reply 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 present an implementation of the methods in the DRAGO reasoning system.

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 Distributed Description Logics

Distributed Description Logics as proposed in [2] provide a language for talking over 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 notion of interpretation introduced above 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 axioms 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 $\mathfrak{J} = (\{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I})$ where $\mathcal{I}_i$ 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 [13]) 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 $\mathfrak{J}$ 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 $\mathfrak{J}$ 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 $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.

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 T-Box can depend on the subsumption between two concepts in a different T-Box 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:

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

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 [13]. To be more specific, adding the inference rule in equation 2 to existing tableaux reasoning methods lead to a correct and complete method for reasoning in DDL. The method has been implemented in the DRAGO system [12] which is available for download at `http://trinity.dit.unitn.it/drago/`.

3 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 [1]. 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.

3.1 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 following two ontologies O_i and O_j with the mappings $\mathcal{B}_{ij}$.

Axioms of O_i	Axioms of O_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 O_i to O_j in $\mathcal{B}_{ij}$
$i:\mathsf{Student} \xrightarrow{\equiv} j:\mathsf{Student}$ $i:\mathsf{PhDStudent} \xrightarrow{\equiv} j:\mathsf{PhDStudent}$ $i:\mathsf{MsStudent} \xrightarrow{\equiv} j:\mathsf{MsStudent}$

It can be easily 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 ontology 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 a consistency for mappings and claim that a mapping is consistent if it does not make a satisfiable concept in the target ontology unsatisfiable:

Definition 1 (Consistency) *A set of bridge rules $\mathcal{B}_{ij}$ from an ontology O_i to an ontology O_j is* consistent *if for all atomic concept C such that $O_j \not\models C \sqsubseteq \bot$, we have that $\langle O_i, O_j, \mathcal{B}_{ij}\rangle \not\models j:C \sqsubseteq \bot$. A set of bridge rules in* inconsistent *if it is 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 rather 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 ontology 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 ontology to the other. For instance, the mapping $i : A \xrightarrow{\sqsubseteq} j : B$ transfer the fact of x being A in ontology O_i, into the fact that $r_{ij}(x)$ is B in O_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) *A set of bridge rules $\mathcal{B}_{ij}$ from O_i to O_j is an* embedding *if for all model $\mathcal{I}_i$ for O_i and for all atomic concepts C of $\mathcal{L}_i$ with $C^{\mathcal{I}_i} \neq \emptyset$, there is a distributed model $\langle \mathcal{I}_i, \mathcal{I}_j, r_{ij} \rangle$ such that $r_{ij}(C^{\mathcal{I}_i}) \neq \emptyset$.*

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 ontologies 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 O_j is an ontology that speaks about food and contains the axioms

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

If ontology O_i speaks about the things sold in a big super-store, which sells food, computers and plants, then you will contains 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 ontology is supposed not to have anything that corresponds to a rose.

To accommodate with this case we can refine the definition of embedding by adding a set of concepts on which we require the domain relation to be defined.

Definition 3 (Embedding for a concept) *A set of bridge rules from O_i to O_j is an* embedding for an atomic concept C *if the following assertion holds: if $O_i \models C$ (C is consistent in O_i) then there is a distributed interpretation $\langle \mathcal{I}_i, \mathcal{I}_j, r_{ij} \rangle$ such that $r(C^{\mathcal{I}_i}) \neq \emptyset$. A set of bridge rules is an embedding for a set of atomic concepts $\mathcal{C}$, if it is an embedding for all the atomic concepts $C \in \mathcal{C}$.*

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 ontologies. Computing the set of non-embedded concepts gives us an idea of topics on which two ontologies 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.

3.2 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 O_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 O_i. In the following definition we formalize the notion of entailment (or consequence) between mappings

Definition 4 (Entailment) *A bridge rule $i : A \xrightarrow{R} j : B$ (with $R \in \{\sqsubseteq, \sqsupseteq, \perp, \equiv\}$) is a* consequence *of a set of bridge rules $\mathcal{B}_{ij}$ from O_i to O_J, if for every distributed model $\mathfrak{I} = \langle \mathcal{I}_i, \mathcal{I}_j, r_{ij} \rangle$ of $\langle O_i, O_j, \mathcal{B}_{ij} \rangle$, $\mathfrak{I} \models i : A \xrightarrow{R} j : B$*

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.

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) *A set of bridge rules $\mathcal{B}_{ij}$ is* contained *in another set of bridge rules $\mathcal{B}'_{ij}$ if and only if for each bridge rule $b \in \mathcal{B}_{ij}$ b is a consequence of $\mathcal{B}'_{ij}$. A set of bridge rules $\mathcal{B}_{ij}$ is* minimal*, if there is no subset S of $\mathcal{B}_{ij}$ such that $\mathcal{B}_{ij}$ is contained in S.*

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.

4 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 $O_i \models C \sqsubseteq \bot$, and $\mathfrak{T} \models i\!:\!C \sqsubseteq \bot$
- *Local/global subsumption:* check if $O_i \models C \sqsubseteq D$, and $\mathfrak{T} \models i\!:\!C \sqsubseteq D$
- *Local/global classification:* Produce a classification[1] on the atomic concepts of O_i.

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.

4.1 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.

[1]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 O_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 $O_i \models C \equiv D$ (resp $\mathfrak{T} \models i: C \equiv D$), $C \prec D$ if and only if $O_i \models C \sqsubseteq D$ and $O_i \not\models C \sqsubseteq D$, (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 $E \in \mathcal{C}$, $C \prec E \prec D$

CONSISTENCYCHECK($\mathfrak{T}_{ij} = \langle O_i, O_j, \mathcal{B}_{ij}\rangle$) computes if the mappings $\mathfrak{B}_{ij}$ are consistent w.r.t. O_i and O_j 1. GLOBALCLASSIFY$_j(\mathfrak{T}_{ij})$ 2. if for some C, such that $C \equiv \bot$, LOCALSAT$_j(C)$ = TRUE then return FALSE else return TRUE

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

4.2 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 ontology.

Definition 6 (Image) *Let $C^{\rightarrow}$ be a concept name that does not appear in O_j and C an atomic concept in O_i. $C^{\rightarrow}$ is called the j-*image *of C in $O_j^{C^{\rightarrow}} := O_j \cup \{C^{\rightarrow} \sqsubseteq \top\}$ if $C^{\rightarrow}$ is defined by the mapping $\mathcal{B}_{ij}^{C^{\rightarrow}} = \mathcal{B}_{ij} \cup \{i:C \xrightarrow{\equiv} j:C^{\rightarrow}\}$.*

For improving the readability we write O_j (resp. $\mathcal{B}_{ij}$) instead of $O_j^{C^{\rightarrow}}$ (resp. $\mathcal{B}_{ij}^{C^{\rightarrow}}$) in the following and assume that for an image $C^{\rightarrow}$ the ontology O_j resp. the set of bridge rules $\mathcal{B}_{ij}$ is already extended.

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 ontology. This means that we can reformulate the embedding property in terms of conditions that only apply to named concepts in the following way.

Proposition 1 (Embedding) *If $O_i \not\models i : C \sqsubseteq \bot$ and $\langle O_i, O_j, \mathcal{B}_{ij}\rangle \not\models j : C^{\rightarrow} \sqsubseteq \bot$ then C is embedded in ontology O_j.*

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

EmbeddingCheck($\langle O_i, O_j, \mathcal{B}_{ij}\rangle, C$) checks if a concept C is embedded in an ontology O_j by the mapping $\mathcal{B}_{ij}$

1. if LocalSat$_i C$ = True and
2. GlobalSat($\langle O_i, O_j, \mathcal{B}_{ij}\rangle, j : C^{\rightarrow}$) = 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 ontology.

4.3 Entailment

The idea of explicitly representing images of concepts in the target ontology can also be used to give an operational definition for testing entailment of bridge rules. This is done by extending the existing mapping with the image of the concept that is the domain of the mapping to be tested and checking the semantic relation between this image and other concepts in the target ontology.

Proposition 2 (Entailment) *Let $C^{\rightarrow}$ be the image of an atomic concept C. For any atomic concept D in ontology O_j then the following equivalences hold:*

$$\begin{array}{lcl}
\langle O_i, O_j, \mathcal{B}_{ij}\rangle \models j : C^{\rightarrow} \equiv D & \iff & \mathfrak{T} \models i : C \xrightarrow{\equiv} j : D \\
\langle O_i, O_j, \mathcal{B}_{ij}\rangle \models j : C^{\rightarrow} \sqsubseteq D & \iff & \mathfrak{T} \models i : C \xrightarrow{\sqsubseteq} j : D \\
\langle O_i, O_j, \mathcal{B}_{ij}\rangle \models j : C^{\rightarrow} \sqsupseteq D & \iff & \mathfrak{T} \models i : C \xrightarrow{\sqsupseteq} j : D \\
\langle O_i, O_j, \mathcal{B}_{ij}\rangle \models j : C^{\rightarrow} \sqcap D \sqsubseteq \bot & \iff & \mathfrak{T} \models i : C \xrightarrow{\bot} j : D
\end{array}$$

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 $i : C \xrightarrow{R} j : Dk$) is a consequence of mapping $\mathcal{B}_{ij}$ w.r.t., O_i and O_j.

1. extend $\mathcal{B}_{ij}$ with $C : i \xrightarrow{\equiv} C^{\rightarrow} : j$ for some new concept $C^{\rightarrow}$ in O_j
2. Case:

R = " $\sqsubseteq$ " return GlobalSubsumption($j : C^{\rightarrow} \sqsubseteq D$)
R = " $\sqsupseteq$ " return GlobalSubsumption($j : D \sqsubseteq C^{\rightarrow}$)
R = " $\equiv$ " return GlobalSubsumption($j : D \sqsubseteq C^{\rightarrow}$) $\wedge$ GlobalSubsumption($j : C^{\rightarrow} \sqsubseteq D$)
R = "$\bot$" return $\neg$GlobalSatisfiable($j : D \sqcap C^{\rightarrow}$)

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.

5 Discussion

We discussed the problem of reasoning about ontology mappings, a problem that has not been studied intensively so far. We identified a number of formal properties that are relevant for judging the quality of a mapping and showed how these properties can be decided based on existing reasoning services for distributed description logics.

So far our work has mainly been theoretical. Next important steps are the implementation of the reasoning services proposed in this paper and their evaluation on the basis of automatically generated mappings between description logic models. In this context, we will have to think about optimized algorithms that avoid exhaustive testing and scales up to real world ontologies.

Another direction for future work is the integration of reasoning about the properties of mappings with automatic mapping methods. We think that the properties defined in this paper can help to prune the search space of automatic matching tools and increase the quality of the generated mappings.

References

[1] 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.

[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 *Second International Semantic Web Conference ISWC'03*, volume 2870 of *LNCS*, pages 164–179. Springer, 2003.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] H. Stuckenschmidt and M. Klein. Integrity and change in modular ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'03*, Aapulco, Mexico, 2003.