

Reconciling concepts and relations in heterogeneous ontologies

Chiara Ghidini and Luciano Serafini

ITC-IRST
Via Sommarive 18
I-38040 Trento, Italy
{ghidini,serafini}@itc.it

Abstract. In the extensive usage of ontologies envisaged by the Semantic Web there is a compelling need for expressing mappings between the components of heterogeneous ontologies. These mappings are of many different forms and involve the different components of ontologies. State of the art languages for ontology mapping enable to express semantic relations between homogeneous components of different ontologies, namely they allow to map concepts into concepts, individuals into individuals, and properties into properties. Many real cases, however, highlight the necessity to establish semantic relations between heterogeneous components. For example to map a concept into a relation or vice versa. To support the interoperability of ontologies we need therefore to enrich mapping languages with constructs for the representation of *heterogeneous mappings*. In this paper, we propose an extension of Distributed Description Logics (DDL) to allow for the representation of mapping between concepts and relations. We provide a semantics of the proposed language and show its main logical properties.

1 Introduction

In the extensive usage of ontologies envisaged by the Semantic Web there is a compelling need for expressing mappings between different and heterogeneous ontologies. These mappings are of many different forms and involve the different components of ontologies.

Most of the formalisms for distributed ontology integration, which are based on the p2p architecture [12], provide a language to express semantic relations between concepts belonging to different ontologies. These classes of languages are usually called *mapping languages* [11, 10]. These formalisms can express that a concept, say *MarriedMan*, in Ontology 1 is equivalent to the concept *Husband* in Ontology 2, or that the concept *Benedict* in Ontology 3 is more specific than the concept *Relative* in Ontology 4. Few mapping languages allow also to express semantic relations between properties in different ontologies (see [7, 5]). However, to the best of our knowledge, none of the existing approaches support mappings between properties and concepts. Such mappings are necessary to express the semantic relations between two ontologies, when the information represented as a concept in the former is represented as a relation in the

latter, or vice versa. As a practical example consider two ontologies. The first one is the ontology <http://www.daml.org/2001/01/gedcom/gedcom> which contains the concept `Family` and the property `spouseIn`. `Family` represents the set of families, and `spouseIn` relates a Human with the family in which he/she is one of the spouses. The second one is the ontology <http://ontologyportal.org/translations/SUMO.owl>, which contains the relation `spouse`, which represents the relationship of marriage between two Humans. In integrating these two ontologies one would like to state, for instance, that every family in the first ontology can be mapped into a married couple in the second ontology, or in other words, that that the concept `Family` can be mapped into the relation `spouse`.

The goal of this paper is to extend a language for ontology mapping, and to introduce mechanisms for the representation of *heterogeneous mappings* between ontologies. We focus on the mappings between concepts and relations as they provide a challenging example of ontology mismatch, we describe their formal semantics and we study their main properties. We adopt the formal framework of Distributed Description Logics (DDL) [9] because it is a formalism which is explicitly constructed to state expressive relations between heterogeneous languages and domains of interpretation. Summarizing, the claimed contributions of this paper are: (i) an expressive mapping language for heterogeneous ontologies; (ii) a clear semantics for the proposed mapping language, and (iii) an investigation of its basic logical properties.

The paper is structured as follows: in Section 2 we motivate our work with a detailed example. In Section 3 we provide an extension of DDL to represent heterogeneous mappings. In Section 4 we study the main properties of the proposed logic. We end with related work (Section 5) and some final remarks (Section 6).

2 A Motivating Example

Let us consider two ontologies from the web. The first is an extensive ontology describing the domain of Geography developed by a corporation¹, the second is the DAML+OIL representation of the 2001 CIA World Fact Book². We call the first Ontology 1, and the second Ontology 2. Looking at the ontologies in detail we have noticed that both need to represent the notion of geographic coordinates. Figures 1 and 2 report the definition of geographic coordinates from the two ontologies.

While the two ontologies are interested in describing the geographic coordinate system of Earth, they have specific views on how to describe this domain of knowledge. Rather than giving a detailed formalization of the example, we focus on the key elements that are affected by the representation of geographical coordinates in the two ontologies. Ontology 1 does not contain an explicit notion of position on earth (or geographical coordinate), and expresses positions in terms of `Latitude` and `Longitude` (see Figure 1). Ontology 2 takes a different perspective and expresses geographical coordinates as a specific concept `LatLon`,

¹ See <http://reliant.tekknowledge.com/DAML/Geography.daml>

² See <http://www.daml.org/2001/12/factbook/factbook-ont>

```

<rdfs:Class rdf:ID= "Latitude">
  <rdfs:subClassOf
    rdf:resource = "http://reliant.tekknowledge.com/DAML/SUMO.owl#Region"/>
  <rdfs:label>latitude</rdfs:label>
  <rdfs:label>parallel</rdfs:label>
  <rdfs:comment>Latitude is the class of Regions,
    associated with areas on the Earth's surface, which are parallels
    measured in PlaneAngleDegrees from the Equator.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID= "Longitude">
  <rdfs:subClassOf
    rdf:resource = "http://reliant.tekknowledge.com/DAML/SUMO.owl#Region"/>
  <rdfs:label>longitude</rdfs:label>
  <rdfs:label>meridian</rdfs:label>
  <rdfs:comment>Longitude is the class of Regions, associated with areas
    on the Earth's surface, which are meridians measured in PlaneAngleDegrees
    from the PrimeMeridian through GreenwichEnglandUK.</rdfs:comment>
</rdfs:Class>

```

Fig. 1. Position = Latitude + Longitude

```

<owl:Class rdf:ID="LatLon">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#latitude"/>
      <owl:allValuesFrom rdf:resource="&xsd;double"/>
      <factbook:units>degrees</factbook:units>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#longitude"/>
      <owl:allValuesFrom rdf:resource="&xsd;double"/>
      <factbook:units>degrees</factbook:units>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Fig. 2. Position = Object with two properties: Latitude and Longitude

which has two properties represented by the roles `latitude` and `longitude` which are numbers of type `Double` (see Figure 2). A graphical representation of two different representations of geographical coordinates inspired by the definitions in Figures 1 and 2 is given in Figure 3. Despite the different choices made by the ontology designers, there is clear relation between Ontology 1 and Ontology 2, and in particular between the concepts `Latitude` and `Longitude` and the properties (roles) `latitude` and `longitude`, respectively.

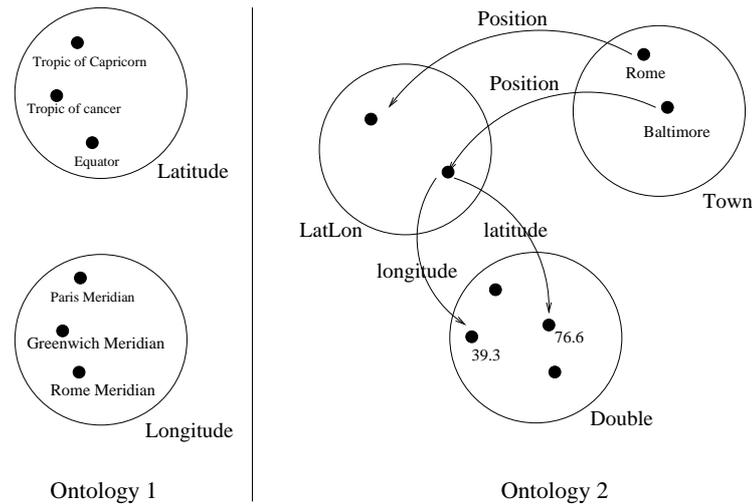


Fig. 3. Representing geographical positions

State of the art formalisms for the representation of distributed ontologies and reasoning, would do very little with this example, except perhaps identifying that `Latitude` and `Longitude` in Ontology 1 are related with `Double` in Ontology 2 (assuming that `Latitude` and `Longitude` are represented as doubles also in Ontology 1). But this is an hardly informative mapping, as it does not capture the essential fact that both ontologies are describing the geographic coordinate system of Earth.

3 An expressive mapping language

Description Logic (DL) has been advocated as the suitable formal tool to represent and reason about ontologies. Distributed Description Logic (DDL) [3, 9] is a *natural* generalization of the DL framework designed to formalize multiple ontologies interconnected by semantic mappings. As defined in [3, 9], *Distributed Description Logic* provides a syntactical and semantical framework for formalization of multiple ontologies *pairwise* linked by semantic mappings. In DDL,

ontologies correspond to description logic theories (T-boxes), while semantic mappings correspond to collections of *bridge rules* (\mathfrak{B}).

In the following we recall the basic definitions of DDL as defined in [9], and we extend the set of bridge rules, introducing new semantic mappings between distributed ontologies.

3.1 Distributed Description Logics: the syntax

Given a non empty set I of indexes, used to identify ontologies, let $\{\mathcal{DL}_i\}_{i \in I}$ be a collection of description logics³. For each $i \in I$ let us denote a T-box of \mathcal{DL}_i as \mathcal{T}_i . In this paper, we assume that each \mathcal{DL}_i is description logic weaker or at most equivalent to \mathcal{SHIQ} . Thus a T-box will contain all the information necessary to define the terminology of a domain, including not just concept and role definitions, but also general axioms relating descriptions, as well as declarations such as the transitivity of certain roles.

We call $\mathbf{T} = \{\mathcal{T}_i\}_{i \in I}$ a family of T-Boxes indexed by I . Intuitively, \mathcal{T}_i is the description logic formalization of the i -th ontology. To make every description distinct, we will prefix it with the index of ontology it belongs to. For instance, the concept C that occurs in the i -th ontology is denoted as $i : C$. Similarly, $i : C \sqsubseteq D$ denotes the fact that the axiom $C \sqsubseteq D$ is being considered in the i -th ontology.

Semantic mappings between different ontologies are expressed via collections of *bridge rules*. In the following we use A and B as placeholders for concepts and R and S as placeholders for relations.

Definition 1 (Bridge rules). *A bridge rule from i to j is an expression defined as follows:*

$$i : A \xrightarrow{\sqsupseteq} j : B \quad (\text{concept-onto-concept bridge rule}) \quad (1)$$

$$i : A \xrightarrow{\sqsubseteq} j : B \quad (\text{concept-into-concept bridge rule}) \quad (2)$$

$$i : R \xrightarrow{\sqsupseteq} j : S \quad (\text{role-onto-role bridge rule}) \quad (3)$$

$$i : R \xrightarrow{\sqsubseteq} j : S \quad (\text{role-into-role bridge rule}) \quad (4)$$

$$i : A \xrightarrow{\sqsupseteq} j : R \quad (\text{concept-onto-role bridge rule}) \quad (5)$$

$$i : A \xrightarrow{\sqsubseteq} j : R \quad (\text{concept-into-role bridge rule}) \quad (6)$$

$$i : R \xrightarrow{\sqsupseteq} j : A \quad (\text{role-onto-concept bridge rule}) \quad (7)$$

$$i : R \xrightarrow{\sqsubseteq} j : A \quad (\text{role-into-concept bridge rule}) \quad (8)$$

where A and B are concepts of \mathcal{DL}_i and \mathcal{DL}_j respectively, and R and S are roles of \mathcal{DL}_i and \mathcal{DL}_j respectively. Bridge rules (1)–(4) are called homogeneous bridge rules, and bridge rules (5)–(8) are called heterogeneous bridge rules

³ We assume familiarity with Description Logic and related reasoning systems, described in [1].

Bridge rules do not represent semantic relations stated from an external *objective* point of view. Indeed, there is no such global view in the web. Instead, bridge rules from i to j express relations between i and j viewed from the *subjective* point of view of the j -th ontology. Let us discuss the different mapping categories.

Homogeneous bridge rules Bridge rules (1) and (2) have been introduced and studied in [3, 9] with the name of *onto-bridge rule* and *into-bridge rule*, respectively. Intuitively, the concept-into-concept bridge rule $i : A \xrightarrow{\sqsubseteq} j : B$ states that, from the j -th point of view the concept A in i is less general than its local concept B . Similarly, the concept-onto-concept bridge rule $i : A \xrightarrow{\supseteq} j : B$ expresses the fact that, according to j , A in i is more general than B in j . Therefore, bridge rules from i to j provide the possibility of translating into j 's ontology (under some approximation) the concepts of a foreign i 's ontology. Note, that since bridge rules reflect a subjective point of view, bridge rules from j to i are not necessarily the inverse of the rules from i to j , and in fact bridge rules from i to j do not force the existence of bridge rules in the opposite direction. Thus, the bridge rule

$$i : \text{Article} \xrightarrow{\supseteq} j : \text{ConferencePaper} \quad (9)$$

expresses the fact that, according to ontology j , the concept `Article` in ontology i is more general than its local concept `ConferencePapers`, while the bridge rules

$$i : \text{Article} \xrightarrow{\sqsubseteq} j : \text{Article} \quad (10)$$

$$i : \text{Article} \xrightarrow{\supseteq} j : \text{Article} \quad (11)$$

say that, according to ontology j , the concept `Article` in ontology j is equivalent to its local concept `Article`. Bridge rules (3) and (4) formalize the analogous intuition for roles. For example, the bridge rule:

$$i : \text{marriedTo} \xrightarrow{\sqsubseteq} j : \text{partnerOf}$$

says that according to ontology j , the relation `marriedTo` in ontology i is less general than its own relation `partnerOf`.

Heterogeneous bridge rules Bridge rules (5) and (6) define how concepts are mapped into roles. Bridge rule (5) states that from the point of view of j concept A in Ontology i corresponds to its own relation R and A is less general than R . Bridge rule (6), on the contrary, states that A is more general than R . For instance, the bridge rule:

$$1 : \text{Latitude} \xrightarrow{\supseteq} 2 : \text{latitude}$$

says that according to ontology 2, concept `Latitude` in ontology 1 is more general than its own relation `latitude`. That is all latitudes in its own ontology have a corresponding `Latitude` in ontology 1.

Bridge rules (7) and (8) define how roles are mapped into concepts, and are the counterpart of bridge rules (5) and (6). Bridge rule (7) says that from the point of view of j role A in Ontology i corresponds to its own concept A and R is less general than A . Bridge rule (8), on the contrary, states that R is more general than A . For example, the bridge rule:

$$1 : \text{spouse} \xrightarrow{\sqsubseteq} 2 : \text{Family}$$

states that every married couple in ontology 1, can be mapped into a family in ontology 2. Similarly the bridge rule.

$$1 : \text{WorksFor} \xrightarrow{\supseteq} 2 : \text{WorkingContract}$$

states that every working contract in ontology 2 corresponds to some working relation in ontology 1.

Bridge rules (5)–(8) are important examples of heterogeneous mappings between ontologies, but the list of heterogeneous bridge rules presented in this paper is by no means complete. We have chosen to study the mappings between relations and concepts as they are a clear and interesting example of heterogeneous mappings. To address the problem of ontology mapping and alignment in full, other forms of heterogeneous mappings need to be investigated, among them mappings between individuals and concepts and even more complex mappings involving interconnected parts of different ontologies.

Definition 2 (Distributed T-box). *A distributed T-box (DTB)*

$$\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$$

consists of a collection $\{\mathcal{T}_i\}_{i \in I}$ of T-boxes, and a collection $\mathfrak{B} = \{\mathfrak{B}_{ij}\}_{i \neq j \in I}$ of bridge rules between them.

3.2 Distributed Description Logics: the semantics

The semantic of DDL, which is a customization of Local Models Semantics [6, 7], assigns to each ontology \mathcal{T}_i a *local interpretation domain*. The first component of an interpretation of a DTB is a family of interpretations $\{\mathcal{T}_i\}_{i \in I}$, one for each T-box \mathcal{T}_i . Each \mathcal{T}_i is called a *local interpretation* and consists of a *possibly empty* domain $\Delta^{\mathcal{T}_i}$ and a valuation function $\cdot^{\mathcal{T}_i}$, which maps every concept to a subset of $\Delta^{\mathcal{T}_i}$, and every role to a subset of $\Delta^{\mathcal{T}_i} \times \Delta^{\mathcal{T}_i}$. The interpretation on the empty domain is denoted with the apex ϵ .

Notice that, in DL, interpretations are defined always on a non empty domain. Therefore \mathcal{I}^ϵ is not an interpretation in DL. In DDL however we need to provide a semantics for *partially inconsistent* distributed T-boxes, i.e. DTBs in which some of the local T-boxes are inconsistent. \mathcal{I}^ϵ provides an “impossible interpretation” which can be associated to inconsistent T-boxes. Indeed, \mathcal{I}^ϵ satisfies every axiom $X \sqsubseteq Y$ (also $\top \sqsubseteq \perp$) since $X^{\mathcal{I}^\epsilon} = \emptyset$ for every concept and role X .

The second component of the DDL semantic are families of domain relations. Domain relations define how the different T-box interact and are necessary to define the satisfiability of bridge rules.

Definition 3 (Domain relation). A domain relation r_{ij} from $\Delta^{\mathcal{I}_i}$ to $\Delta^{\mathcal{I}_j}$ is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$. We use $r_{ij}(d)$ to denote $\{d' \in \Delta^{\mathcal{I}_j} \mid \langle d, d' \rangle \in r_{ij}\}$; for any subset D of $\Delta^{\mathcal{I}_i}$, we use $r_{ij}(D)$ to denote $\bigcup_{d \in D} r_{ij}(d)$; for any $R \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ we use $r_{ij}(R)$ to denote $\bigcup_{\langle d, d' \rangle \in R} r_{ij}(d) \times r_{ij}(d')$.

A domain relation r_{ij} represents a possible way of mapping the elements of $\Delta^{\mathcal{I}_i}$ into its domain $\Delta^{\mathcal{I}_j}$, seen from j 's perspective. For instance, if $\Delta^{\mathcal{I}_1}$ and $\Delta^{\mathcal{I}_2}$ are the representation of time as Rationals and as Naturals, r_{ij} could be the round off function, or some other approximation relation. This function has to be conservative w.r.t., the order relations defined on Rationals and Naturals. Domain relation is used to interpret homogeneous bridge rules according with the following definition.

Definition 4 (Satisfiability of homogeneous bridge rules). The domain relation r_{ij} satisfies a homogeneous bridge rule w.r.t., \mathcal{I}_i and \mathcal{I}_j , in symbols $\langle \mathcal{I}_i, r_{ij}, \mathcal{I}_j \rangle \models br$, according with the following definition:

1. $\langle \mathcal{I}_i, r_{ij}, \mathcal{I}_j \rangle \models i : A \xrightarrow{\sqsubseteq} j : B$, if $r_{ij}(A^{\mathcal{I}_i}) \subseteq B^{\mathcal{I}_j}$
2. $\langle \mathcal{I}_i, r_{ij}, \mathcal{I}_j \rangle \models i : A \xrightarrow{\supseteq} j : B$, if $r_{ij}(A^{\mathcal{I}_i}) \supseteq B^{\mathcal{I}_j}$

where A and B are either two concept expressions or two role expressions.

Domain relations do not provide sufficient information to evaluate the satisfiability of heterogeneous mappings. Intuitively, an heterogeneous bridge rule between a relation R and a concept A connects a pair of objects related by R with an object which is in A . This suggests that, to evaluate heterogeneous bridge rules from roles in i to concepts in j one needs a relation that maps pair of objects in $\Delta^{\mathcal{I}_i}$ into objects of $\Delta^{\mathcal{I}_j}$, and to evaluate a heterogeneous bridge rule from concepts in i to roles in j one needs a relation that maps objects in $\Delta^{\mathcal{I}_i}$ into pairs of objects in $\Delta^{\mathcal{I}_j}$.

Definition 5 (Concept-role and role-concept domain relation). A concept-role domain relation cr_{ij} from $\Delta^{\mathcal{I}_i}$ to $\Delta^{\mathcal{I}_j}$ is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$. A role-concept domain relation rc_{ij} from $\Delta^{\mathcal{I}_i}$ to $\Delta^{\mathcal{I}_j}$ is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j} \times \Delta^{\mathcal{I}_j}$.

We use $cr_{ij}(d)$ to denote $\{\langle d_1, d_2 \rangle \in \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i} \mid \langle d, d_1, d_2 \rangle \in cr_{ij}\}$; for any subset D of $\Delta^{\mathcal{I}_i}$, we use $cr_{ij}(D)$ to denote $\bigcup_{d \in D} cr_{ij}(d)$. We use $rc_{ij}(\langle d_1, d_2 \rangle)$ to denote $\{d \in \Delta^{\mathcal{I}_i} \mid \langle d_1, d_2, d \rangle \in rc_{ij}\}$; for any subset R of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$, we use $rc_{ij}(R)$ to denote $\bigcup_{\langle d_1, d_2 \rangle \in R} rc_{ij}(\langle d_1, d_2 \rangle)$.

Domain relation cr_{ij} represents a possible way of mapping elements of $\Delta^{\mathcal{I}_i}$ into pairs of elements in $\Delta^{\mathcal{I}_j}$, seen from j 's perspective. For instance, if $\Delta^{\mathcal{I}_1}$ and $\Delta^{\mathcal{I}_2}$ are the representation of geographical coordinates as in Figure 4, cr_{12} could

be the function mapping latitude values into the corresponding latitudes. For instance, by setting

$$cr_{12}(\text{TropicOfCancer}^{\mathcal{I}_1}) = \{\langle x, 23.27 \rangle \in \text{latitude}^{\mathcal{I}_2}\}$$

we can represent the fact that the tropic of cancer is associated with pairs of objects $\langle x, y \rangle$ such that y is the latitude of x and y is equal to 23.27 (the latitude of the tropic of cancer). Vice-versa a domain relation rc_{ij} represents a possible way of mapping a pair of $\Delta^{\mathcal{I}_i}$ into the corresponding element in $\Delta^{\mathcal{I}_j}$. For instance, if the pair $\langle \text{John}^{\mathcal{I}_1}, \text{Mary}^{\mathcal{I}_1} \rangle \in \text{spouse}^{\mathcal{I}_1}$, then the fact that

$$rc_{12}(\text{John}^{\mathcal{I}_1}, \text{Mary}^{\mathcal{I}_1}) = \text{family23}^{\mathcal{I}_2}$$

represents the fact that family23 is the family containing the married couple of John and Mary.

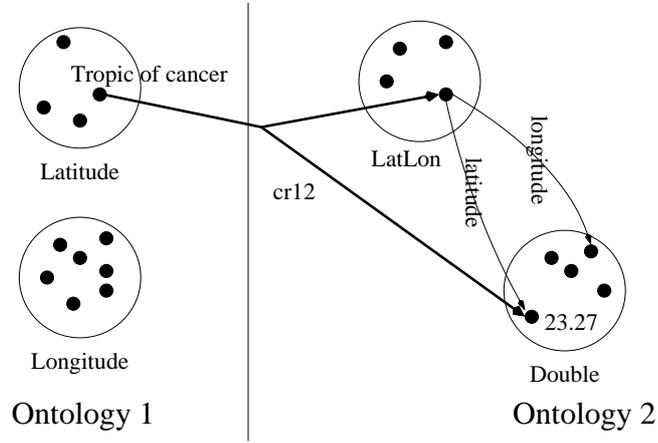


Fig. 4. Concept-role domain relation

Definition 6 (Satisfiability of heterogeneous bridge rules). *The concept-role domain relation cr_{ij} satisfies a concept to role bridge rule w.r.t., \mathcal{I}_i and \mathcal{I}_j , in symbols $\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \models br$, according with the following definition:*

1. $\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \models i : A \xrightarrow{\sqsubseteq} j : R$, if $cr_{ij}(A^{\mathcal{I}_i}) \subseteq R^{\mathcal{I}_j}$
2. $\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle \models i : A \xrightarrow{\sqsupseteq} j : R$, if $cr_{ij}(A^{\mathcal{I}_i}) \supseteq R^{\mathcal{I}_j}$

where A is a concept expression of i and R a role expression of j .

The role-concept domain relation rc_{ij} satisfies a role to concept bridge rule w.r.t., \mathcal{I}_i and \mathcal{I}_j , in symbols $\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle \models br$, according with the following definition:

1. $\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle \models i : R \xrightarrow{\sqsubseteq} j : A$, if $rc_{ij}(R^{\mathcal{I}_i}) \subseteq A^{\mathcal{I}_j}$

2. $\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle \models i : R \xrightarrow{\exists} j : A$, if $rc_{ij}(R^{\mathcal{I}_i}) \supseteq A^{\mathcal{I}_j}$

where A is a concept expression of j and R a role expression of i .

Definition 7 (Distributed interpretation). A distributed interpretation

$$\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I}, \{cr_{ij}\}_{i \neq j \in I}, \{rc_{ij}\}_{i \neq j \in I} \rangle$$

of a DTB \mathfrak{T} consists of local interpretations \mathcal{I}_i for each \mathcal{T}_i on local domains $\Delta^{\mathcal{I}_i}$, and families of domain relations r_{ij} , cr_{ij} and rc_{ij} between these local domains.

Definition 8 (Satisfiability of a Distributed T-box). A distributed interpretation \mathfrak{I} satisfies the elements of a DTB \mathfrak{T} according to the following clauses: for every $i, j \in I$

1. $\mathfrak{I} \models i : A \sqsubseteq B$, if $\mathcal{I}_i \models A \sqsubseteq B$
2. $\mathfrak{I} \models \mathcal{T}_i$, if $\mathfrak{I} \models i : A \sqsubseteq B$ for all $A \sqsubseteq B$ in \mathcal{T}_i
3. $\mathfrak{I} \models \mathfrak{B}_{ij}$, if
 - $\langle \mathcal{I}_i, r_{ij}, \mathcal{I}_j \rangle$ satisfies all the homogeneous bridge rules in \mathfrak{B}_{ij} ,
 - $\langle \mathcal{I}_i, cr_{ij}, \mathcal{I}_j \rangle$ satisfies all the concept-to-role bridge rules in \mathfrak{B}_{ij} ,
 - $\langle \mathcal{I}_i, rc_{ij}, \mathcal{I}_j \rangle$ satisfies all the role-to-concept bridge rules in \mathfrak{B}_{ij}
4. $\mathfrak{I} \models \mathfrak{T}$, if for every $i, j \in I$, $\mathfrak{I} \models \mathcal{T}_i$ and $\mathfrak{I} \models \mathfrak{B}_{ij}$

Definition 9 (Distributed Entailment and Satisfiability). $\mathfrak{T} \models i : C \sqsubseteq D$ (read as “ \mathfrak{T} entails $i : C \sqsubseteq D$ ”) if for every \mathfrak{I} , $\mathfrak{I} \models \mathfrak{T}$ implies $\mathfrak{I} \models_d i : C \sqsubseteq D$. \mathfrak{T} is satisfiable if there exists a \mathfrak{I} such that $\mathfrak{I} \models \mathfrak{T}$. Concept $i : C$ is satisfiable with respect to \mathfrak{T} if there is a \mathfrak{I} such that $\mathfrak{I} \models \mathfrak{T}$ and $C^{\mathcal{I}_i} \neq \emptyset$.

4 Characterizing mappings

In this section we enunciate the most important properties of the extended version of DDL and for each result we provide an example that explains why this property is desirable. We assume familiarity with Description Logics, and in particular with *SHIQ*. Symbols, \sqcup , \sqcap , and \neg denote the usual union, intersection, and inverse operators of Description Logics. Similarly, $\exists R.C$ is used to denote the existential restriction.

Theorem 1 (General property). If $\mathfrak{I} \models i : A \xrightarrow{\exists} j : G$, and $\mathfrak{I} \models i : B \xrightarrow{\sqsubseteq} j : H$, then

$$\mathfrak{I} \models i : A \sqsubseteq B \implies \mathfrak{I} \models j : G \sqsubseteq H \quad (12)$$

where the pair A and B and the pair G and H are pairs of homogeneous elements, that is either pairs of concepts or pairs of roles.

Example 1. Let

- $\mathfrak{I} \models i : \text{Article} \xrightarrow{\exists} j : \text{ConferencePaper}$, and
- $\mathfrak{I} \models i : \text{Article} \xrightarrow{\sqsubseteq} j : \text{ScientificArticle}$.

Theorem 1 allows to infer that a conference paper is a Scientific article in O_j , namely $\mathcal{I} \models j : \text{ConferencePaper} \sqsubseteq \text{Article}$, from the fact that $\mathcal{I} \models i : \text{Article} \sqsubseteq \text{Article}$. Similarly, let

- $\mathcal{I} \models i : \text{Couple} \xrightarrow{\exists} j : \text{partnerOf}$, and
- $\mathcal{I} \models i : \text{Family} \xrightarrow{\sqsubseteq} j : \text{spouseIn} \sqcup \text{childIn}$,

where Couple and Family are concepts and partnerOf, and spouseIn relations. If $\mathcal{I} \models i : \text{Couple} \sqsubseteq \text{Family}$, then $\mathcal{I} \models j : \text{partnerOf} \sqsubseteq \text{spouseIn} \sqcup \text{childIn}$.

Theorem 2 (Concept into/onto concept). *If $\mathcal{I} \models i : A \xrightarrow{\exists} j : G$, and $\mathcal{I} \models i : B_k \xrightarrow{\sqsubseteq} j : H_k$ for $1 \leq k \leq n$ (with $n \geq 0$), then:*

$$\mathcal{I} \models i : A \sqsubseteq \bigsqcup_{k=1}^n B_k \implies \mathcal{I} \models j : G \sqsubseteq \bigsqcup_{k=1}^n H_k \quad (13)$$

where A , G , B_k and H_k ($1 \leq k \leq n$) are concepts.

Example 2. Let

- $\mathcal{I} \models i : \text{ArchivalPublication} \xrightarrow{\exists} j : \text{ArchivalPublication}$,
- $\mathcal{I} \models i : \text{ConferencePaper} \xrightarrow{\sqsubseteq} j : \text{Article}$, and
- $\mathcal{I} \models i : \text{BookChapter} \xrightarrow{\sqsubseteq} j : \text{Article}$.

Theorem 3 guarantees that if $\mathcal{I} \models i : \text{ArchivalPublication} \sqsubseteq \text{ConferencePaper} \sqcup \text{BookChapter}$ then $\mathcal{I} \models j : \text{ArchivalPublication} \sqsubseteq \text{Article}$.

Theorem 3 (Concept into/onto role). *If $\mathcal{I} \models i : A \xrightarrow{\exists} j : R$, and $\mathcal{I} \models i : B_k \xrightarrow{\sqsubseteq} j : S_k$ for $1 \leq k \leq n$ (with $n \geq 0$), then:*

$$\mathcal{I} \models i : A \sqsubseteq \bigsqcup_{k=1}^n B_k \implies \mathcal{I} \models j : \exists R.X \sqsubseteq \bigsqcup_{k=1}^n \exists S_k.X \quad (14)$$

where A and B_k ($1 \leq k \leq n$) are concepts, R and S_k ($1 \leq k \leq n$) are roles, and X is any arbitrary concept.

Example 3. Let

- $\mathcal{I} \models i : \text{Parent} \xrightarrow{\exists} j : \text{hasChild}$,
- $\mathcal{I} \models i : \text{Mother} \xrightarrow{\sqsubseteq} j : \text{motherOf}$, and
- $\mathcal{I} \models i : \text{Father} \xrightarrow{\sqsubseteq} j : \text{fatherOf}$, and

where Parent, Mother and Father are concepts, while hasChild, motherOf and fatherOf are relations. If $\mathcal{I} \models i : \text{Parent} \sqsubseteq \text{Mother} \sqcup \text{Father}$, then $\mathcal{I} \models j : \exists \text{hasChild}.X \sqsubseteq \exists \text{motherOf}.X \sqcup \exists \text{fatherOf}.X$, for any concept X .

Theorem 4 (Role into/onto role).

$$\text{If } \mathcal{I} \models i : R \xrightarrow{\sqsubseteq} j : S, \text{ then } \mathcal{I} \models i : R^- \xrightarrow{\sqsubseteq} j : S^- \quad (15)$$

$$\text{If } \mathcal{I} \models i : R \xrightarrow{\sqsupseteq} j : S, \text{ then } \mathcal{I} \models i : R^- \xrightarrow{\sqsupseteq} j : S^-, \quad (16)$$

Example 4. Let $\mathcal{I} \models i : \text{marriedTo} \xrightarrow{\sqsupseteq} j : \text{partnerOf}$, then $\mathcal{I} \models i : \text{marriedTo}^- \xrightarrow{\sqsupseteq} j : \text{partnerOf}^-$. Similarly for the into case.

Corollary 1 (Role into/onto concept). *If $\mathcal{I} \models i : R \xrightarrow{\sqsupseteq} j : A$, and $\mathcal{I} \models i : S \xrightarrow{\sqsubseteq} j : B_k$ for $1 \leq k \leq n$ (with $n \geq 0$), then:*

$$\mathcal{I} \models i : R \sqsubseteq S \implies \mathcal{I} \models j : A \sqsubseteq \bigsqcup_{k=1}^n B_k \quad (17)$$

$$\mathcal{I} \models i : R \sqsubseteq S \implies \mathcal{I} \models j : A \sqsubseteq \bigsqcap_{k=1}^n B_k \quad (18)$$

where R , and S are roles and A and B_k ($1 \leq k \leq n$) are concepts.

Example 5. Let

- $\mathcal{I} \models i : \text{holdsSeasonalTicketOf} \xrightarrow{\sqsupseteq} j : \text{SeasonalTicketHolder}$
- $\mathcal{I} \models i : \text{supporterOf} \xrightarrow{\sqsubseteq} j : \text{Person}$, and
- $\mathcal{I} \models i : \text{supporterOf} \xrightarrow{\sqsubseteq} j : \text{JuventusFan}$

where $\text{holdsSeasonalTicketOf}$, Person and JuventusFan are concepts, while $\text{SeasonalTicketHolder}$, supporterOf and JuventusFan are relations. If

$$\mathcal{I} \models i : \text{holdsSeasonalTicketOf} \sqsubseteq \text{supporterOf}$$

then

$$\mathcal{I} \models i : \text{SeasonalTicketHolder} \sqsubseteq \text{Person} \sqcup \text{JuventusFan}$$

and

$$\mathcal{I} \models i : \text{SeasonalTicketHolder} \sqsubseteq \text{Person} \sqcap \text{JuventusFan}$$

Corollary 2 (concept into/onto role). *If $\mathcal{I} \models i : A \xrightarrow{\sqsupseteq} j : R$, and $\mathcal{I} \models i : B \xrightarrow{\sqsubseteq} j : S$, then:*

$$\mathcal{I} \models i : A \sqsubseteq B \implies \mathcal{I} \models j : R^- \sqsubseteq S^- \quad (19)$$

where A , and B are concepts and R and S are roles.

Example 6. Let

- $\mathcal{I} \models i : \text{Mother} \xrightarrow{\sqsupseteq} j : \text{motherOf}$
- $\mathcal{I} \models i : \text{Parent} \xrightarrow{\sqsubseteq} j : \text{hasChild}$,

where `Parent`, and `Mother` are concepts, while `hasChild`, and `motherOf` are relations. If $\mathcal{I} \models i : \text{Mother} \sqsubseteq \text{Parent}$, then $\mathcal{I} \models j : \text{motherOf}^- \sqsubseteq \text{hasChild}^-$.

Corollary 3 (role into/onto role). *If $\mathcal{I} \models i : R \xrightarrow{\sqsupseteq} j : P$, and $\mathcal{I} \models i : S \xrightarrow{\sqsubseteq} j : Q$, then:*

$$\mathcal{I} \models i : R \sqsubseteq S^- \implies \mathcal{I} \models j : P \sqsubseteq Q^- \quad (20)$$

where R , S , P and Q are roles.

Example 7. Let

- $\mathcal{I} \models i : \text{marriedTo} \xrightarrow{\sqsupseteq} j : \text{partnerOf}$, and
- $\mathcal{I} \models i : \text{livingWith} \xrightarrow{\sqsupseteq} j : \text{friendOf}$.

If $\mathcal{I} \models i : \text{marriedTo} \sqsubseteq \text{livingWith}^-$, then $\mathcal{I} \models j : \text{partnerOf} \sqsubseteq \text{friendOf}^-$.

Theorem 5 (Role union). *If the DL includes role union $R \sqcup S$, then Theorem 2 can be generalised to all bridge rules, with the only constraint of A and B_k and G and H_k families of homogeneous elements, that is either families of concepts or families of roles.*

5 Related Work

All the mapping languages described in [11] do not support full heterogeneous mappings. In general, however, mapping languages support a limited version of heterogeneous mappings. For instance, in [5], it is possible to express the mapping

$$\forall x. (\exists y. R_1(x, y) \rightarrow C_2(x)) \quad (21)$$

or, similarly, in the original version of DDL one can state the mapping

$$1 : \exists R. \top \xrightarrow{\sqsubseteq} 2 : C \quad (22)$$

However, the encoding of heterogeneous mappings shown above is not very expressive and its usage can also lead to undesirable consequences. For instance, assume a relation `IsMarried` exists in ontology 1, and a concept `Marriage` exists in ontology 2. Assume we want to impose that the relation `IsMarried` in ontology 1 is equivalent to the concept `Marriage` in ontology 2, and we only have mappings as in Equation (22). Then, we can only state mappings of the form:

$$\begin{aligned} 1 : \exists \text{IsMarried}. \top &\xrightarrow{\sqsubseteq} 2 : \text{Marriage} \\ 1 : \exists \text{IsMarried}. \top &\xrightarrow{\sqsupseteq} 2 : \text{Marriage} \end{aligned}$$

But these mappings express something rather different from our initial goal as they map single elements of a couple into marriages. Moreover, assume we also

have a bridge rule mappings wives in ontology 1 into women in ontology 2 as follows:

$$1 : \text{Wife} \xrightarrow{\sqsubseteq} 2 : \text{Woman}$$

together with the axiom

$$\text{Wife} \sqsubseteq \exists \text{IsMarried}.\top$$

in ontology 1 stating that a wife is a married entity. From all this we can infer in ontology 2 that a wife is a marriage, i.e.,

$$\text{Wife} \sqsubseteq \text{Marriage}$$

This undesirable conclusion reflects the fact that in mapping the two ontologies, we have identified the participants of a relation, (the married person) with the relation itself (the marriage). To avoid this bad behavior, Omelayenko claims in [8] that mappings between classes and properties are not relevant from an application point of view. We believe that the examples shown in the paper provide a convincing evidence that this is not the case, and that an appropriate formalization of heterogeneous mappings can avoid some of the problems mentioned above.

An effort towards the formalization of heterogeneous mappings between concepts and relations in the area of federated databases is described in [2]. In this work the authors define five types of correspondences between concepts and properties. If A is a concept and R is a relation, they consider the following correspondences:

- A is equivalent to R ;
- A is more general to R ;
- A is less general to R ;
- A and R do overlap;
- A and R do not overlap.

The semantics of the correspondences above can be expressed by the following mappings:

- $\forall x.(A(x) \leftrightarrow \exists y.R(y, x))$;
- $\forall x.(\exists y.R(y, x) \rightarrow A(x))$;
- $\forall x.(A(x) \rightarrow \exists y.R(y, x))$;
- $\exists x.(A(x) \wedge \exists y.R(y, x))$;
- $\forall x.(A(x) \rightarrow \neg \exists y.R(y, x))$.

This semantics is similar to the encoding described in Equation (21). The only difference is that it considers the range of the relation R in place of the domain. Therefore it suffers of problems similar to the ones shown above for Equation (21).

Different forms of mappings (bridge rules) have been studied in other formalisms strictly related to DDL, such as C-OWL [4] and DFOL [7]. Both formalisms do not address the problem of heterogeneous mappings and should therefore be extended in this direction.

6 Concluding Remarks

The language and the semantics presented in this paper constitute a genuine contribution in the direction of the integration of heterogeneous ontologies. The language proposed in this paper makes it possible to directly bind a concept with a relation in a different ontology, and vice-versa. At the semantic level we have introduced a domain relation that maps pairs of object into objects and vice-versa. This also constitute a novelty in the semantics of knowledge integration. We have showed the main formal properties of the mapping language, and we have left the complete characterization of the logic for future work.

References

1. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. J. M. Blanco, A. Illarramendi, and A. Goñi. Building a federated database system: An approach using a knowledge base system. *International Journal of Intelligent and Cooperative Information Systems*, 3(4):415–455, 1994.
3. A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003. LNCS 2800, Springer Verlag.
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 Verlag, 2003.
5. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *23rd ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS 2004)*, pages 241–251, 2004.
6. C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.
7. C. Ghidini and L. Serafini. Distributed First Order Logics. In *Frontiers Of Combining Systems 2, Studies in Logic and Computation*, pages 121–140. Research Studies Press, 1998.
8. B. Omelayenko. Integrating vocabularies: Discovering and representing vocabulary maps. In *Proceedings of the First International Semantic Web Conference (ISWC-2002)*, volume 2342 of LNCS. Springer-Verlag, 2003.
9. L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In *19th Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 576–581, 2005.
10. Luciano Serafini, Alex Borgida, and Andrei Tamilin. Aspects of distributed and modular ontology reasoning. In *19th Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 570–575, 2005.
11. H. Stuckenschmidt and M. Uschold. Representation of semantic mappings. In *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005.
12. Ilya Zaihrayeu and Matteo Bonifacio, editors. *Proceedings of the First Workshop on Peer-to-Peer Knowledge Management (P2PKM 2004)*, volume 108 of CEUR Workshop Proceedings. CEUR-WS.org, 2004.