

Three Semantics for Distributed Systems and their Relations with Alignment Composition*

Antoine Zimmermann¹ and Jérôme Euzenat¹

INRIA Rhône-Alpes
Montbonnot Saint-Martin, France
{Antoine.Zimmermann, Jerome.Euzenat}@inrialpes.fr

Abstract. An ontology alignment explicitly describes the relations holding between two ontologies. A system composed of ontologies and alignments interconnecting them is here in called a distributed system. We give three different semantics of a distributed system, that do not interfere with the semantics of ontologies. Their advantages are compared, with respect to allowing consistent merge of ontologies, managing heterogeneity and complying with an alignment composition operation. We show that only the two first variants, which differ from other proposed semantics, can offer a sound composition operation.

1 Introduction

In a general sense, ontology alignment is an explicit description of the semantic relationship that exists between different ontologies. However, in several practical applications, it is restricted to a set of syntactical correspondences or mapping. For instance, the Alignment API [1] defines a correspondence as a pair of entities (one from each ontology), together with the type of relation, and the confidence in its correctness. The API output format has been used in several ontology matching tools, and our goal is to consider alternative semantics for this generic type of alignments.

We define three different semantics that have advantages and drawbacks. The first one, *simple distributed semantics*, considers the whole distributed system as a coherent knowledge representation which can be interpreted in a single domain. It is appropriate for consistently merging ontologies, based on correspondences. However, it is not tolerant to any inconsistency, be it local or global. In very heterogeneous systems, like the semantic web or semantic P2P systems, this feature is not desirable. So we extend the first semantics to *integrated distributed semantics* where each local knowledge representation is interpreted in its own domain but these interpretation are then correlated in a global domain. Finally, we define a *contextualized distributed semantics*, inspired by DFOL/DDL/C-OWL [2–4], where there is no global domain of interpretation: each local ontologies “imports” knowledge from other ontologies in its own context.

* This work is partly supported by the Knowledge Web NoE (IST-2004-507482).

Additionally, the semantics is parameterized by the set of expressible relations between ontology entities. It means that this semantics is usable to reason about class relations (*e.g.*, equivalence, subsumption, disjunction), as well as temporal relations or spatial relations, *etc.*

Finally, we study the semantics of an alignment composition operator and prove that only the first two semantics comply with it.

This paper is articulated as follows: Sect. 2 gives an overview of previously proposed semantics for schema mappings, ontology alignments and matching, or distributed knowledge representation. Sect. 3 describes the syntax of distributed systems. Sect. 4 gives the simple distributed semantics. Sect. 5.1 extends the semantics to integrated distributed semantics, through the use of an *equalizing function* to a global domain. Sect. 5.2 presents the contextualized variant of the semantics. Sect. 6 discusses the composition of alignments within each of the proposed semantics.

2 Related Work

Ontology matching is strongly related to database schema matching. However, as noted in [5], there are foundational differences between schema matching and ontology matching. The most prominent being the weakness of schemata semantics. As a result, the schema mapping community is concerned about *query answering*, while ontology alignment must offer more general *reasoning* capabilities in distributed systems. Of particular interest is [6], which develops a notion of satisfied mappings that is very generic and the approach in this paper is very similar to theirs. The difference resides in their using of “helper models”, and they do not provide a composition operator. Such operator is provided by [7] for relational databases, and cannot be exploited as it is for more general ontology alignment composition. Other database-related work about composition comprises [8–10].

Another different approach to the semantics of schema mapping and alignments is found in [11] and generalized in [12] where the authors use the theory of institutions [13] to characterize the relation between models of related ontologies or schemata. The approach is seducing for our ontology alignment purpose: it allows a language independent definition of ontology relation, elegant constructions of merging, composing and it is grounded on model theory. Nonetheless, it has a major problem: the “ontology morphisms” can only account for the most basic relations between ontological entities, namely equivalence. So we consider such abstraction to be inappropriate for more general types of relations [14].

Other semantics for distributed knowledge based systems have been proposed. Various distributed logics are found in [2] (DFOL), [3] (DDL), [4] (C-OWL). They all assume that each ontology is interpreted within a context that has to be related to others in order to interpret the whole system. This approach effectively solves the problem of mutual inconsistencies in heterogeneous knowledge representations, but we show that contextualizing the semantics forbids a sound composition operation.

3 Syntax

We want to design a model-theoretic semantics for distributed systems, which are composed of (1) ontologies and (2) alignments. The present section discusses the syntax of those components.

One of the key features of our alignment semantics is its independence from the ontology language. Sometimes, we will use the term “element of an ontology” to refer to any syntactical entity mentioned in the ontology, *e.g.*, class, property, individual. The only restriction upon the choice of the ontology language is the existence of a model-theoretic semantics. Among such languages, we can cite RDF [15], OWL [16], FOL, Conceptual Graphs [17], *etc.*

We follow the definition of alignment found in [18], but we intentionally discard the confidence value.¹

Definition 1 (Ontology element). *An ontology element is either a term of the ontology (e.g., class, property, individual) or a compound entity built out of other ontology elements and constructors from the ontological language.*

Definition 2 (Ontology element relation). *An ontology element relation R is a symbol denoting a binary relation or predicate \tilde{R} . Given a specific alignment language, the set of usable relation symbols is written \mathfrak{R} .*

The relations an alignment language can capture may be: equivalence, subsumption, disjunction of classes; temporal/spatial relations; spatial relations; fuzzy membership; *etc.*

These definitions makes the components of a correspondence.

Definition 3 (Correspondence). *A correspondence is a triple $\langle e_1, e_2, R \rangle$ where:*

- e_1 and e_2 are ontology elements from the two ontologies to align;
- R is an ontology element relation that is asserted to hold between e_1 and e_2 .

Example 1. $\langle \text{Man1} \sqcup \text{Woman1}, \text{Human2}, \equiv \rangle$ or $\langle \text{Girl1}, \text{Human2} \sqcap \text{Female2}, \sqsubseteq \rangle$ are examples of correspondences.

Definition 4 (Ontology alignment). *An ontology alignment between ontologies O_1 and O_2 is a set of correspondences with ontology elements belonging to O_1 and O_2 .*

The structure that our semantics interprets is composed of multiple ontologies and alignments between them. It is called a distributed system.

Definition 5 (Distributed system). *A distributed system (DS for short) is a pair $\langle (O_i), (A_{ij}) \rangle$ consisting of a family of ontologies $(O_i)_{i \in I}$ over a set of indexes I interconnected by a family of alignments $(A_{ij})_{i, j \in I}$.*

Although there can be pairs of ontologies that are not connected by an alignment, we will then consider the missing alignment as an empty set of correspondences. So A_{ij} is always defined.

¹ How to treat uncertainty in this semantics is still a subject of investigations.

4 Simple Distributed Semantics

The simple distributed semantics consider a distributed system (DS) as a coherent knowledge base. This means that all ontologies are interpreted within the same domain. We first give the definitions for local semantics (given by the ontological language), then our proposed DS semantics follows.

4.1 Local Semantics

The semantics of a DS depends on the semantics of the ontology language. In fact, given a set of ontologies and a set of alignments between them, we can evaluate the semantics of the whole system in function of the semantics of each individual ontology.

Definition 6 (Interpretation of an ontology). *Given an ontology O , an interpretation of O is a function m from elements of O to elements of a domain of interpretation D .*

This is a very general notion of interpretation. Among interpretations, there are particular ones that are said to *satisfy* the ontology. The local semantics of ontologies determine the satisfaction relation \models that relates interpretations to satisfied ontologies, *i.e.*, $m \models O$ iff m satisfies O . An ontology is satisfied if all of its axioms are satisfied. For instance, `A subclass of B` would be satisfied iff A is interpreted as a subset of the interpretation of B . The collection of interpretations that satisfy O (the models of O) is written $\text{Mod}(O)$.

4.2 Satisfaction of an Alignment

In order to determine the semantics of a DS, we first define when a pair of local interpretations satisfies a correspondence and an alignment.

Definition 7 (Satisfied correspondence). *Let O_1, O_2 be two ontologies and $c = \langle e_1, e_2, R \rangle$ be a correspondence between O_1 and O_2 . c is satisfied by interpretations m_1, m_2 of O_1, O_2 iff $m_1(e_1) \tilde{R} m_2(e_2)$. This is written $m_1, m_2 \models c$.*

The relation symbol R is out of the ontology languages. So it does not have to be interpreted in the local semantics. For instance, a temporal relation can be expressed between two OWL classes. The associated relation \tilde{R} is fixed, given a set of relation \mathfrak{R} . For instance, relation symbol \equiv could be associated to the relation “=” (equality) over sets.

If all correspondences are satisfied, then it is said that the pair of interpretations is a model of the alignment.

Definition 8 (Model of an alignment). *A model of an alignment A between ontologies O_1 and O_2 is a pair m_1, m_2 of interpretations of O_1, O_2 such that for all $c \in A$, $m_1, m_2 \models c$. It is noted $m_1, m_2 \models A$.*

The models of an alignment do not take the semantics of the ontologies into account. They just consider the internal consistency of the correspondences. This is interesting because we can reason with and about alignments without actually accessing the aligned ontologies. The collection of all models of an alignment A is written $\text{Mod}(A)$.

Correspondences play the role of axioms that constrain the satisfying interpretations of the aligned ontologies. They therefore act as interpretation constraints of the distributed system.

4.3 Models of a DS

In this section, I denotes a set of indexes. Informally, interpretations of a DS are tuples of local interpretations.

Definition 9 (Interpretation of a DS). *An interpretation of a DS $\langle(O_i), (A_{ij})\rangle$ is a family (m_i) of local interpretations over a common domain D such that for all $i \in I$, m_i is an interpretation of O_i .*

Among interpretations, some are said to *satisfy* the DS. In order to satisfy a DS, interpretations must satisfy constraints given by (1) the ontologies axioms and (2) the alignments correspondences.

Definition 10 (Model of a DS). *A model of a DS $\mathcal{S} = \langle(O_i), (A_{ij})\rangle$ is an interpretation (m_i) of \mathcal{S} such that:*

- $\forall i \in I, m_i \in \text{Mod}(O_i)$ (i.e., m_i is a (local) model of O_i);
- $\forall i, j \in I, i \neq j, m_i, m_j \models A_{ij}$.

This is written $(m_i) \models_s \mathcal{S}$. If a model exists for \mathcal{S} , we say that \mathcal{S} is satisfiable.

We can see that this definition employs a very global view of the models. All ontologies and alignments are taken into account at the same time, and there are strong interdependencies. This is because the DS is seen as a single theory, with ontologies being but mere modules.

However, it is often the case when we only want to reason about local data, while taking advantage of external knowledge. So we define local models modulo a DS:

Definition 11 (Local models modulo a DS). *Local models of an ontology O_i modulo \mathcal{S} are the local models $\text{Mod}_{\mathcal{S}}(O_i) = \{m_i \in \text{Mod}(O_i); \exists (m_j)_{j \neq i} \in \text{Mod}(O_j), (m_i)_{i \in I} \models \mathcal{S}\}$. It corresponds to the projection of the models of a DS on the i th component.*

With this definition, the models of the full system must be known to compute the local models. In order to build more efficient reasoners, we define another notion of models that do not require total integration of all ontologies and alignments at once. It is based on an iterative process of gradually reducing the local models.

Definition 12 (Models of an ontology modulo alignment). Given an ontology O_1 aligned with O_2 according to alignment A , the models of O_1 modulo A are those models of O_1 that satisfy each element of A :

$$\text{Mod}_A(O_1) = \{m_1 \in \text{Mod}(O_1); \exists m_2 \in \text{Mod}(O_2); m_1, m_2 \models A\}$$

Models modulo alignment is the first step of the following iterative definition.

Definition 13 (Iterated local models modulo a DS). Given a DS $\mathcal{S} = \langle (O_i), (A_{ij}) \rangle$, consider $\text{Mod}_{\mathcal{S}}^0(O_i) = \text{Mod}(O_i)$, and the following iterative definition:

$$\text{Mod}_{\mathcal{S}}^k(O_i) = \{m_i \in \text{Mod}_{\mathcal{S}}^{k-1}(O_i); \forall j \in I \setminus \{i\}, \exists m_j \in \text{Mod}_{\mathcal{S}}^{k-1}(O_j); m_i, m_j \models A_{ij}\}$$

$\overline{\text{Mod}}_{\mathcal{S}}(O)$ denotes the limit of the sequence $(\text{Mod}_{\mathcal{S}}^n(O))$ when $n \rightarrow \infty$, i.e., $\overline{\text{Mod}}_{\mathcal{S}}(O) = \text{Mod}_{\mathcal{S}}^{\infty}(O)$.

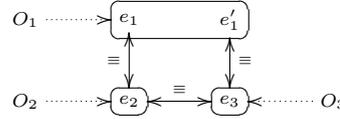
Definition 14 (Local satisfiability). A DS \mathcal{S} is locally satisfiable iff for each ontology O , $\overline{\text{Mod}}_{\mathcal{S}}(O) \neq \emptyset$.

So, the iterated models only give a local view of what happens in the DS. Moreover, the stepwise restriction of the models allows for a faster but approximate reasoning while trying to find new ontologies and alignments.

Proposition 1. Let \mathcal{S} be a DS and O an ontology of \mathcal{S} . For all $n \in \mathbb{N}$, $\text{Mod}_{\mathcal{S}}(O) \subseteq \overline{\text{Mod}}_{\mathcal{S}}(O) \subseteq \text{Mod}_{\mathcal{S}}^{n+1}(O) \subseteq \text{Mod}_{\mathcal{S}}^n(O)$.

Proposition 2. There exists a DS \mathcal{S} such that $\text{Mod}_{\mathcal{S}}(O) \neq \overline{\text{Mod}}_{\mathcal{S}}(O)$.

Proof. We give a sketch of the proof² with a diagram representing the DS.



In this DS, we have $\text{Mod}_{\mathcal{S}}^n(O_i) = \text{Mod}(O_i)$ for all $n \in \mathbb{N}$ and $i \in \{1, 2, 3\}$. But $\text{Mod}_{\mathcal{S}}(O_1)$ is restricted to the models of O_1 where e_1 and e_1' are interpreted as the same entity. \square

In spite of this unfortunate property, $\text{Mod}_{\mathcal{S}}(O)$ and $\overline{\text{Mod}}_{\mathcal{S}}(O)$ are two solutions to the fixed-point equation $\overline{\text{Mod}}_{\mathcal{S}}(O_i) = \{m \in \text{Mod}(O_i); \forall j \in I \setminus \{i\}, \exists m_j \in \overline{\text{Mod}}_{\mathcal{S}}(O_j); m_i, m_j \models A_{ij}\}$. This means that locally reasoning with iterated models will not contradict neighborhood reasoning.

The proposed semantics is somewhat strict, with regard to heterogeneous systems, because it only allows to assert a correspondence when it is fully compatible with both ontologies. While it may be desirable in a few applications, this semantics is not adapted to every ontology alignment use cases. For instance, in the semantic web, ontologies will vary tremendously in size, scope, scale, point of view and quality. We consider two semantics that address this problem.

² For a detailed proof of this property, please refer to <http://www.inrialpes.fr/exmo/people/zimmer/ISWC2006proof.pdf>.

5 Dealing with Heterogeneous Domains

In very heterogeneous applications, having a unified interpretation of a distributed system is not feasible. To address this issue, we propose two variants of the primary semantics: (1) in the *integrated distributed semantics*, local interpretation domains are separated, and they are reconciled in a global domain, thanks to the use of *equalizing functions* (Def. 15) that act as filters; (2) in the *contextualized distributed semantics*, no global domain exists, but interpretation domains are interrelated by as many translation function as there are pairs of domains.

5.1 Integrated Distributed Semantics

The choice of the interpretation domain is not only guided by the interpreter, but also partly decided by the local language semantics. So we will use the concept of an equalizing function to help making the domain commensurate.

Definition 15 (Equalizing function). *Given an interpretation $(m_i)_{i \in I}$ of a DS, an equalizing function $(\gamma_i)_{i \in I}$ over $(m_i)_{i \in I}$ is a family of functions from the local domains of interpretation of (m_i) to a global domain \mathcal{U} .*

So equalizing functions not only define a global domain for the interpretation of the DS, but also define how local domains are correlated in the global interpretation.

Definition 16 (Integrated interpretation of a DS). *An integrated interpretation of a DS \mathcal{S} is a pair $\langle (m_i), \gamma \rangle$ where (m_i) is a simple interpretation of \mathcal{S} and γ is an equalizing functions over (m_i) .*

The integrated interpretations that satisfy the DS are given by the following definition.

Definition 17 (Integrated model of a DS). *An integrated interpretation $\langle (m_i), \gamma \rangle$ of a DS \mathcal{S} is an integrated model iff $\forall i, j \in I, \gamma_i m_i, \gamma_j m_j \models A_{ij}$ and m_i is a local model of O_i .*

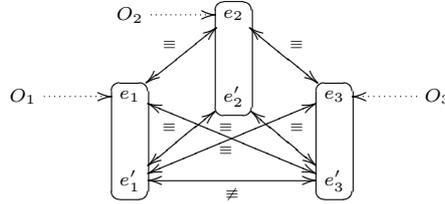
We can define the iterated models of a DS in the following way:

Definition 18 (Integrated iterated local models modulo a DS). *Given a D.S $\mathcal{S} = \langle (O_i), (A_{ij}) \rangle$, consider $\text{Mod}_{\mathcal{S}}^0(O_i) = \text{Mod}(O_i)$, and the following iterative definition:*

$$\begin{aligned} \text{Mod}_{\mathcal{S}}^k(O_i) = \\ \{m_i \in \text{Mod}_{\mathcal{S}}^{k-1}(O_i); \forall j \in I \setminus \{i\}, \exists m_j \in \text{Mod}_{\mathcal{S}}^{k-1}(O_j), \exists \gamma; \gamma_i m_i, \gamma_j m_j \models A_{ij}\} \end{aligned}$$

As with simple distributed semantics, there is a notion of local and global satisfiability (see Def. 14). The integrated iterated models have the same property as the simple iterated models (Prop. 1 and Prop. 2).

Proof (of Prop. 2). We give a sketch of the proof³ with a diagram representing the DS.



This system is locally satisfiable but not globally satisfiable. \square

For particular applications, it may prove useful to give restrictions on equalizing functions. For instance, it might be needed to have injective functions, or to have inclusion-preserving functions. Although we do not describe the treatment of such restrictions, they should be compared to domain relation constraints in DFOL [2]. However, the approach in DFOL differs, because no global domain exists. They rather relates the local domains to each others, allowing to contextualize import of knowledge from and to each DS node. We extend our semantics in order to match this approach.

5.2 Contextualized Distributed Semantics

The contextualized semantics uses domain relation instead of equalizing functions. Domain relations differ from equalizing function in the fact that there exists one function for each pair of ontologies and they relate two local interpretation domains.

Definition 19 (Domain relation). *Given two domains of interpretation D_1 and D_2 , a domain relation is a mapping $r_{12} : D_1 \rightarrow D_2$.*

These domain relations form a part of a contextualized interpretation.

Definition 20 (Contextualized interpretation of a DS). *A contextualized interpretation of a DS \mathcal{S} is a pair $\langle (m_i), (r_{ij}) \rangle$ where (m_i) is a simple interpretation of \mathcal{S} and $(r_{ij})_{i \neq j}$ is a family of domain relations such that r_{ij} relates the domain of m_i to the domain of m_j .*

The models in the contextualized semantics are defined as follows.

Definition 21 (Contextualized model of a DS). *A contextualized interpretation $\langle (m_i), (r_{ij}) \rangle$ of a DS \mathcal{S} is a contextualized model iff $\forall i, j \in I, r_{ij}m_i, m_j \models A_{ij}$ and each m_i is a local model of O_i .*

We again define the iterative models of a DS in the following way:

³ For a detailed proof of this property, please refer to <http://www.inrialpes.fr/exmo/people/zimmer/ISWC2006proof.pdf>.

Remark 1. “;” may also be given by a table of composition. In that case, relations $R \in \mathfrak{R}$ are sets of primitive relations. Moreover, composition is associative iff “;” is associative.

In our first semantic approach, the models of A are pairs of interpretations of O_1 and O_2 , so $\text{Mod}(A_{12})$ is a set-theoretic relation. Relations are composable, and ideally the composition of A_{12} and A_{23} should have equal models as the composition of $\text{Mod}(A_{12})$ and $\text{Mod}(A_{23})$.

Let \mathcal{S} be a DS having 3 ontologies O_1, O_2, O_3 and 2 alignments A_{12}, A_{23} .

Definition 24 (Simple semantic composition). *The simple semantic composition of the simple models of A_{12} and A_{23} , noted $\text{Mod}(A_{23}) \circ_s \text{Mod}(A_{12})$ is the set:*

$$\{\langle m_1, m_3 \rangle \in \text{Mod}(O_1) \times \text{Mod}(O_3); \exists m_2 \in \text{Mod}(O_2), \langle m_1, m_2, m_3 \rangle \in \text{Mod}(\mathcal{S})\}$$

In the case of the integrated semantics, the definition should include the equalizing function.

Definition 25 (Integrated semantic composition). *The integrated semantic composition of the integrated models of A_{12} and A_{23} , noted $\text{Mod}(A_{23}) \circ_i \text{Mod}(A_{12})$ is the set:*

$$\{\langle \langle m_1, m_3 \rangle, \langle \gamma_1, \gamma_3 \rangle \rangle; \exists m_2, \gamma_2, \langle \langle m_1, m_2, m_3 \rangle, \langle \gamma_1, \gamma_2, \gamma_3 \rangle \rangle \in \text{Mod}(\mathcal{S})\}$$

Similarly, the contextualized semantics define a composition with domain relations.

Definition 26 (Contextualized semantic composition). *The contextualized semantic composition of the contextualized models of A_{12} and A_{23} , noted $\text{Mod}(A_{23}) \circ_c \text{Mod}(A_{12})$ is the set:*

$$\{\langle \langle m_1, m_3 \rangle, \langle r_{13}, r_{31} \rangle \rangle; \exists m_2, r_{12}, r_{21}, r_{23}, r_{32}, \langle (m_i)_{i \in \{1,2,3\}}, (r_{ij})_{i \neq j} \rangle \in \text{Mod}(\mathcal{S})\}$$

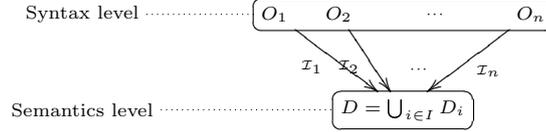
These definitions are rather intuitive and correspond to what is found in constraint reasoning literature, with slight variants due to the presence of equalizing functions and domain relations. The following section compare the three ontologies, and shows that composition is semantically sound in the first two semantics, but not in the contextualized one.

7 Comparing Semantics

Our three semantics do not only differ by their conceptual design. They also imply technical differences.

7.1 Simple Semantics

The following diagram helps visualize the idea behind the simple semantics. Each ontology is treated as a module of a bigger ontology, interpreted in a single domain.



This semantics offer several advantages. It may be used as a general semantics for modules composition in ontology engineering. In practical cases, modules are generally related to each others with an import directive, and sometimes additional bridge axioms are added in the ontological language. With this semantics, modules can be written in different languages and aligned with yet another alignment language. Besides, the decision procedure is easier than in the other distributed semantics. Consider the case of OWL⁵ ontologies with possible relations being `subclassOf`, `equivalentClass`, `disjointClass`, `subPropertyOf`, `equivalentProperty`, then reasoning will not differ from reasoning with a single OWL ontology.

Additionally, composition has the following property:

Property 1. If for all $R_1, R_2 \in \mathfrak{R}$, for all appropriate X, Y, Z , $X \widetilde{R}_1 Y \wedge Y \widetilde{R}_2 Z \Rightarrow X \widetilde{R}_1 ; R_2 Z$, then the simple semantic composition of the models of the alignments is included in the models of the syntactic composition of alignments, *i.e.*, :

$$\text{Mod}(B) \circ \text{Mod}(A) \subseteq \text{Mod}(B \circ A).$$

Proof. Let $m_1, m_3 \in \text{Mod}(B) \circ \text{Mod}(A)$. There exists $m_2 \in \text{Mod}(O_2)$ such that $m_1, m_2 \models A$ and $m_2, m_3 \models B$. Let $c = (e_1, e_3, R)$ be a correspondence of $B \circ A$. There exists $R_1, R_2 \in \mathfrak{R}$ and e_2 in O_2 such that $R = R_1 ; R_2$, $(e_1, e_2, R_1) \in A$ and $(e_2, e_3, R_2) \in B$. We have $m_1(e_1) \widetilde{R}_1 m_2(e_2)$ and $m_2(e_2) \widetilde{R}_2 m_3(e_3)$. The assumption made ensures that $m_1(e_1) \widetilde{R}_1 ; R_2 m_3(e_3)$. So for all $c \in B \circ A$, $m_1, m_3 \models c$. As a result, all m_1, m_3 in $\text{Mod}(B) \circ \text{Mod}(A)$ are also in $\text{Mod}(B \circ A)$. \square

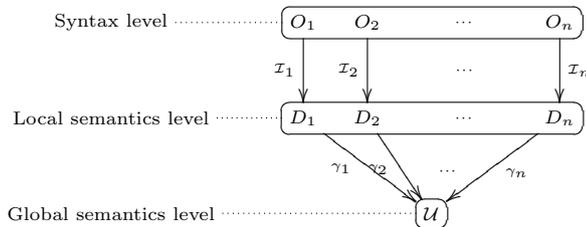
The property required ($\forall R_1, R_2 \in \mathfrak{R}, \forall X, Y, Z, X \widetilde{R}_1 Y \wedge Y \widetilde{R}_2 Z \Rightarrow X \widetilde{R}_1 ; R_2 Z$) is really associated to the common sense meaning of what must be a composition operation. This property is mentioned in work on composition table (*e.g.*, [19]) as a basic property for a sound composition operation. This property encourage reuse of alignments by combining them.

Nonetheless, as explained in [2], interpreting a distributed system in a unique domain is only feasible in the least heterogeneous systems.

⁵ <http://www.w3.org/TR/owl-guide/>

7.2 Integrated Semantics

In the integrated semantics, ontologies are interpreted at the local level, and the equalizing function γ serves to coordinate local interpretations in a global domain.



This approach is much more tolerant to inconsistencies, because the equalizing functions act as a filter between local and global interpretations. Having two levels of interpretations allows one to maintain local specificities while integrating knowledge into one consistent interpretation. And, obviously, if a simple distributed model exists, then an integrated model exists.

The following example demonstrates the interest of equalizing function.

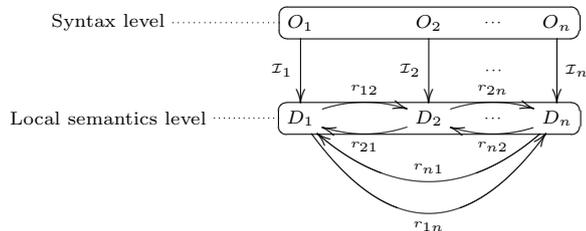
Example 2. Consider two instances of class **Child** that have different ages and different heights, but truly represent the same kid at different moments. In some ontology languages, it may not be possible to interpret these different instances as only one unique individual because age and height may be restricted in cardinality. However, an equalizing function can map two different instances of child with different ages in the local domain, to one instance of a person having no attribute **age** in the global domain.

Moreover, Prop. 1 holds too in this semantics. Therefore, it is also appropriate to manipulate and reuse several alignments in a consistent way. So, this semantics consistently extend the previous one.

However, the reasoning procedure is rendered more difficult because of the presence of equalizing functions. This is quite inevitable since dealing with inconsistencies has always been a time-consuming and complex task.

7.3 Contextualized Semantics

In the contextualized semantics, we drop the global domain, but the domain relations enables each ontology to integrate the whole distributed system in its own context.

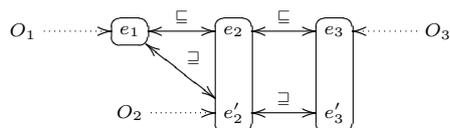


This approach is very similar to context-based logics approach and the interest of contextualizing inferences is explained in *e.g.*, [2].

However, the following result tend to disqualify this semantics when composing alignments becomes a necessity:

Proposition 3. *Prop. 1 does not hold in contextualized semantics.*

Proof. Consider the following DS⁶:



□

Additionally, we show the generality of our integrated semantics with the following proposition:

Proposition 4. *If a contextualized model exists for a DS, then there exists an integrated model.*

Proof. Let $\langle (m_i), (r_{ij}) \rangle$ be a model of a DS. Let $k \in I$ be an indice then $\langle (m_i)_{i \in I}, (r_{ik})_{i \in I} \rangle$, with $r_{kk} = id_{D_k}$ is an integrated model of the DS with global domain D_k . □

8 Conclusion and future work

We presented three variant semantics (*simple*, *integrated* and *contextualized*) for ontology alignments and distributed systems. Our characterization of an alignment allows the definition of an alignment composition operator, which is sound in the first two proposed semantics. We gave the advantages and drawbacks of each approach, and highlighted the benefits of using the first two semantics in comparison to using a contextualized semantics. We also make our approach distinct from others by using a set of relations as a parameter. It enables one to use the same meta-semantics for different types of relations (*e.g.*, temporal/spatial relations, class relations or even data types relations).

We will extend this semantics to include operations such as inverse alignment, union/intersection, and data transformations induced by them (along the line of the schema mapping algebra in [7]). Finally, our semantics and composition operator shall be extended to correspondences with confidence value.

⁶ For a detailed proof of this property, please refer to <http://www.inrialpes.fr/exmo/people/zimmer/ISWC2006proof.pdf>.

References

1. Euzenat, J.: An API for ontology alignment. In: Proc. Third International Semantic Web Conference (ISWC'04). (2004) 698–712
2. Ghidini, C., Serafini, L.: Distributed First Order Logics. In: *Frontiers of Combining Systems 2*. (2000) 121–139
3. Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating information from peer sources. *J. Data Semantics* (2003) 153–184
4. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: Proc. 2nd ISWC. (2003)
5. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: The state of the art. In: *Semantic Interoperability and Integration. Dagstuhl Seminar Proceedings* (2005)
6. Madhavan, J., Bernstein, P.A., Domingos, P., Halevy, A.Y.: Representing and reasoning about mappings between domain models. In: Proc. 18th AAAI'02. (2002) 80–86
7. Bernstein, P.A., Halevy, A.Y., Pottinger, R.A.: A vision for management of complex models. *SIGMOD Record* (2000) 55–63
8. Madhavan, J., Halevy, A.Y.: Composing mappings among data sources. In: Proc. 29th VLDB. (2003) 572–583
9. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Composing schema mappings: Second-order dependencies to the rescue. In: Proc. 23rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems (PODS'04). (2004) 83–94
10. Dragut, E., Lawrence, R.: Composing mappings between schemas using a reference ontology. In: Proc. ODBASE. (2004) 783–800
11. Alagić, S., Bernstein, P.A.: A model theory for generic schema management. In: Proc. 8th DBPL. (2001) 228–246
12. Goguen, J.A.: Data, schema and ontology integration. In: *CombLog workshop*. (2004)
13. Goguen, J.A., Burstall, R.M.: Institutions: abstract model theory for specification and programming. *Journal of the ACM* (1992) 95–146
14. Hitzler, P., Euzenat, J., Krötzsch, M., Serafini, L., Stuckenschmidt, H., Wache, H., Zimmermann, A.: Integrated view and comparison of alignment semantics. Deliverable, Knowledge Web NoE (2006)
15. Hayes, P.: RDF Semantics. Technical report, W3C (2004)
16. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax. Technical report, W3C (2004)
17. Sowa, J.F.: Conceptual graphs for a data base interface. *IBM Journal of Research and Development* **20** (1976) 336–357
18. Bouquet, P., Euzenat, J., Franconi, E., Serafini, L., Stamou, G., Tessaris, S.: Specification of a common framework for characterizing alignment. Deliverable 2.2.1, Knowledge Web NoE (2004)
19. Eschenbach, C.: Viewing composition tables as axiomatic systems. In: Proc. 2nd FOIS. (2001) 93–104