Introduction to ontology matching and alignment (first M2R lecture notes)

Jérôme Euzenat
INRIA Rhône-Alpes & LIG,
Montbonnot, France
Jerome.Euzenat@inrialpes.fr

December 22, 2007

Abstract

These notes are largely extracted from [9].

1 Motivation

We are dealing with a space of knowledge sources called the semantic web. Each source provides knowledge expressed in its own vocabulary called ontology[1]. There is no a priori reason that this knowledge be expressed:

- in the same knowledge representation language,
- with the same vocabulary, or
- with the same modelling of that vocabulary.

Example 1 (Heterogeneity)**.** *Consider two knowledge sources to be found on the the web The first one, o, is a simple set of triples :*

```
May childOf Peter.
Peter childOf Paul.
Paul childOf Zeno.
```

The second source, o', is an axiom in some first order logic dialect:

$$\texttt{grandParent(x, y)} \Leftarrow \exists\, \texttt{z; parent(x, z)} \wedge \texttt{parent(z, y)}$$

One would expect, given these two sources to be able to deduce that `grandParent( Paul, May )` *(that we will refer to as δ holds. This is however not the case since:*

- *The two sources use different knowledge representation languages (syntactic heterogeneity);*

[1] In fact, an ontology provides both the vocabulary and axioms governing the terms in the vocabulary

- *The two sources use different vocabularies:* `child` *versus* `parent` *(terminological heterogeneity);*
- *Even if the vocabulary were the same, the modelling of these notion may be different.*

This can be summarised by the following remarks:

- $o \not\models \delta$ *and* $o' \not\models \delta$*;*
- $o \cup o'$ *impossible*
- *even* $o \cup o' \not\models \delta$

This problem can be solved by imposing to the word a unique ontology and a unique knowledge representation language. However, this is both unrealistic and unwishable. Instead, we propose to draw relations between knowledge sources mostly based on their ontologies. We call "ontology matching" the process of finding the relations between ontologies and we call "alignment" the result of this process expressing declaratively these relations. In an open world in which ontologies evolve, managing ontologies requires using alignments for expressing the relations between ontologies.

Example 2 (Alignment)**.** *In the present case, we can imagine to provide an alignment A containing the following correspondence:*

```
childOf^-1 ≡ parent
```

This alignment would solve our current problem because $o \cup o' \cup A \models \delta$. However, the merge of the ontologies provided by $o \cup o' \cup A$ may not be the solution that works in any case. For instance, if o' contained in addition, the following axioms:

```
parent( x, y ) ⇒ Human( x ) ∧ Human( y )
childOf( x, y ) ⇒ Tree( x ) ∧ Tree( y )
Human ⊥ Tree
```

The merge would be an inconsistent knowledge source. In such a case, it may be better to use the alignment as a data transformation programme that would import whatever `childOf` *formula entailed by o into the corresponding* `parent` *formula in o'.*

Several classes of applications can be considered (they are more extensively described in [9], we only summarise them here). They are the following:

Ontology evolution uses matching for finding the changes that have occurred between two ontology versions [5].

Schema integration uses matching for integrating the schemas of different databases under a single view;

Catalog integration uses matching for offering an integrated access to on-line catalogs;

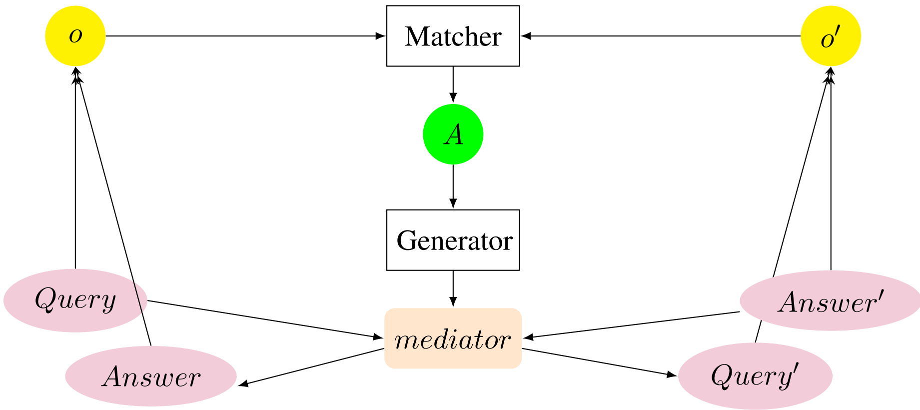


Figure 1: Query mediation (from [9]). From two matched ontologies o and o', resulting in alignment A, a *mediator* is generated. This allows the transformation of queries expressed with the entities of the first ontology into a query using the corresponding entities of a matched ontology and the translation back of the results from the second ontology to the first one.

Data integration uses matching for integrating the content of different databases under a single database;

P2P information sharing uses matching for finding the relations between ontologies used by different peers [21], Figure 1 is an example of the use of alignments for mediation in such systems;

Web service composition uses matching between ontologies describing service interfaces in order to compose web services by connecting their interfaces;

Multiagent communication uses matching for finding the relations between the ontologies used by two agents and translating the messages they exchange;

Context matching in ambient computing uses matching of application needs and context information when applications and devices have been developed independently and use different ontologies;

Query answering uses ontology matching for translating user queries about the web;

Semantic web browsing uses matching for dynamically (while browsing) annotating web pages with partially overlapping ontologies.

It is clear, from the above examples, that matching ontologies is a major issue in ontology related activities. It is not circumscribed to one area of ontology, but applies to any application that communicates through ontologies.

We will be interested here, in providing the basis for ontology matching and alignment within the semantic web. In particular, we will establish the constraints

2 Reminder: ontology semantics

An ontology is a set of assertions that selects the set of interpretations which satisfy them. These interpretations are called models. They constitute the possible interpretations of an ontology.

Definition 1 (Model)**.** *Given an ontology o, a model of o is an interpretation $m = \langle I, D \rangle$ of o, which satisfies all the assertions in o:*

$$\forall \delta \in o, m \models \delta$$

The set of models of an ontology o is denoted as $\mathcal{M}(o)$.

Finally, an important notion is the set of assertions that are consequences of an ontology. These are the assertions implicitly entailed by an ontology and they determine the answers to queries.

Definition 2 (Consequence)**.** *Given an ontology formula δ, δ is a consequence of an ontology o, if and only if, it is satisfied by all models of o. This is denoted as $o \models \delta$.*

Given a model m, we will denote as $m(e)$ the application of the interpretation function of the model to some ontology entity e.

This digression introduced more precisely, albeit generally, a simplified syntax and semantics of ontologies. This will be useful when considering the meaning of matching ontologies.

3 Alignment syntax

Alignments express the correspondences between entities belonging to different ontologies. All definitions here are given for matching between two ontologies. In case of multiple matching, the definitions can be straightforwardly extended by using n-ary correspondences. A correspondence must consider the two corresponding entities and the relation that is supposed to hold between them. We provide the definition of the alignment following the work in [7; 2].

Since the related entities are an important part of alignments, they have to be defined. We separate the matched entities from the ontology language because it can be desirable to have a different language for identifying the matched entities. Given an ontology language, we use an *entity language* for expressing those entities that will be put in correspondence by matching. The expressions of this language will depend on the ontology on which expressions are defined.

The entity language can be simply made of all the formulas of the ontology language based on the ontology vocabulary. It can restrict its scope to particular kinds of formulas from the language, for instance, atomic formulas, or even to terms of the language, like class expressions. It can also restrict the entities to be only named entities. This is convenient in the context of the semantic web to restrict entities to those identifiable by their URIs. The entity language can also be an extension of the ontology language: this can be a query language, such as SPARQL [19], adding operations for manipulating ontology entities that are not available in the ontology language itself, like concatenating strings or joining relations. Finally, this entity language can combine both extension and restriction, e.g., by authorising any boolean operations over named ontology entities.

Definition 3 (Entity language)**.** *Given an ontology language L, an entity language Q_L is a function from any ontology $o \subseteq L$ which defines the matchable entities of ontology o.*

In the following we will assume that each ontology interpretation can be extended to an interpretation of the entity language associated with the ontology.

The next important component of the alignment is the relation that holds between the entities. We identify a set of relations Θ that is used for expressing the relations between the entities. Matching algorithms primarily use the equivalence relation ($=$) meaning that the matched objects are the same or are equivalent if these are formulas. It is possible to use relations from the ontology language within Θ. For instance, using OWL, it is possible to take advantage of the `owl:equivalentClass`, `owl:disjointWith` or `rdfs:subClassOf` relations in order to relate classes of two ontologies. These relations correspond to set-theoretic relations between classes: *equivalence* ($=$); *disjointness* ($\perp$); *more general* ($\sqsupseteq$). They can be used without reference to any ontology language. Finally, relations can be of any type and are not restricted to relations present within the ontology language, such as fuzzy relations or probability distributions over a complete set of relations, or similarity measures.

For pragmatic reasons, the relationship between two entities is assigned a degree of confidence which can be viewed as a measure of trust in the fact that the correspondence holds – 'I trust 70% the fact that the correspondence is correct or reliable' – and can be compared with the certainty measures provided with meteorological forecasts.

Definition 4 (Confidence structure)**.** *A confidence structure is an ordered set of degrees $\langle \Xi, \leq \rangle$ for which there exists a greatest element $\top$ and a smallest element $\bot$.*

The usage of confidence degrees is that the higher the degree with regard to $\leq$, the most likely the relation holds. It is convenient to interpret the greatest element as the boolean true and the smallest element as the boolean false.

The most widely used structure is based on the real number unit interval $[0\ 1]$, but some systems simply use the boolean lattice. Some other possible structures are fuzzy degrees, probabilities or other lattices. [13] has investigated the structure of fuzzy confidence relations. This structure can be extended, for instance, if one wants to compose alignments. Thus, in this case, it may be necessary to define operations for combining these degrees.

With these ingredients, it is possible to define the correspondences that have to be found by matching algorithms.

Definition 5 (Correspondence)**.** *Given two ontologies o and o' with associated entity languages Q_L and $Q_{L'}$, a set of alignment relations Θ and a confidence structure over Ξ, a* correspondence *is a 5-uple:*

$$\langle id, e, e', r, n \rangle,$$

such that

- id *is a unique identifier of the given correspondence;*
- $e \in Q_L(o)$ *and* $e' \in Q'_{L'}(o')$*;*
- $r \in \Theta$*;*
- $n \in \Xi$*.*

The correspondence $\langle id, e, e', r, n \rangle$ asserts that the relation r holds between the ontology entities e and e' with confidence n.

Example 3 (Correspondence)**.** *For example, a simple kind of correspondence is as follows:*

$$\begin{array}{l}\textit{http://book.ontologymatching.org/example/culture-shop.owl\#Book} = \\ \qquad\qquad\textit{http://book.ontologymatching.org/example/library.owl\#Volume}\end{array}$$

It asserts the equivalence relation with full confidence between what is denoted by two URIs, namely the Book *class in one ontology and the* Volume *class in another one. Some examples of more complex correspondences are as follows:*

$$\textit{author}(x, concat(w.\textit{firstname}, w.\textit{lastname})) \Leftarrow_{.85} \begin{array}{l} \quad\textit{Book}(x) \\ \wedge\ \textit{writtenBy}(x, w) \\ \wedge\ \textit{Writer}(w) \end{array}$$

is a Horn clause expressing that if there exists a Book x written by Writer w*, the* author *of x in the first ontology is identified by the concatenation of the first and last name of w. The confidence in this clause is quantified with a .85 degree.*

There can be several possible correspondences for the same entities depending on the language in which correspondences are expressed. For instance, one could have the simple correspondence that speed *in one ontology is equivalent to* velocity *in another one:*

$$\textit{speed} \equiv \textit{velocity}$$

or record that they are expressed in miles per hour and metre per second respectively:

$$\textit{speed} = \textit{velocity} \times 2.237$$
$$0.447 \times \textit{speed} = \textit{velocity}$$

Finally, an alignment is defined as a set of correspondences.

Definition 6 (Alignment)**.** *Given two ontologies o and o', an* alignment *is made up of a set of correspondences between pairs of entities belonging to $Q_L(o)$ and $Q_{L'}(o')$ respectively.*

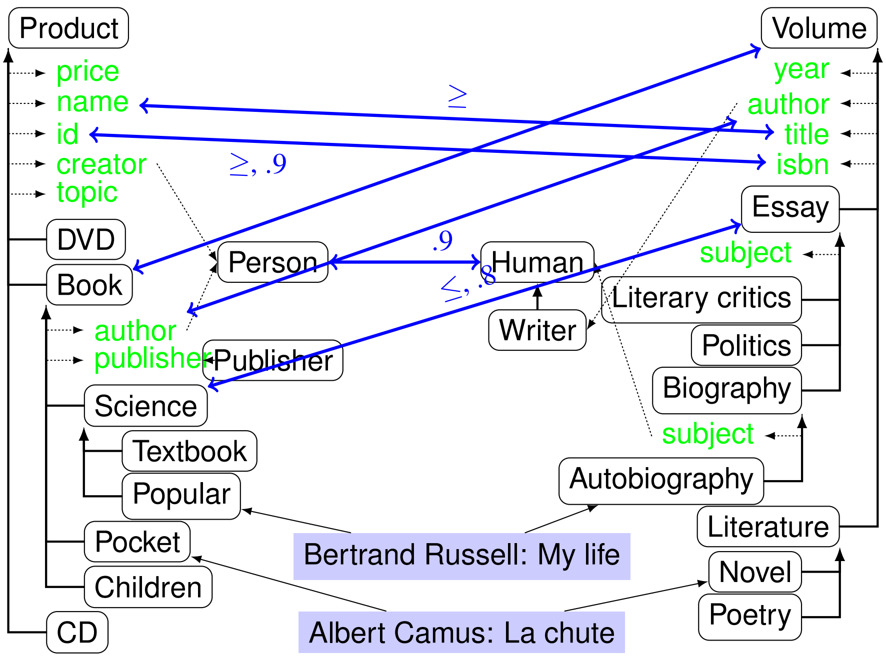


Figure 2: Alignment between two ontologies. Correspondences are expressed by arrows. By default their relation is $=$ and their confidence value is 1.0; otherwise, these are mentioned near the arrows.

Example 4 (Alignment)**.** *Figure 2 displays a possible alignment for a pair of ontologies. It can be expressed by the following correspondences:*

$\textit{Book} =_{1.0} \textit{Volume}$	$\textit{name} \geq_{1.0} \textit{title}$
$\textit{id} \geq_{.9} \textit{isbn}$	$\textit{author} =_{1.0} \textit{author}$
$\textit{Person} =_{.9} \textit{Human}$	$\textit{Science} \leq_{.8} \textit{Essay}$

So far, our alignments are very simple: they are sets of pairs of entities from two ontologies. However, there are, in the literature, at least three types of n:m, multiple or complex alignments:

1. alignments involving more than two ontologies produced by multiple matching, that we may call multialignments,
2. alignments involving correspondences between more than two entities (still belonging to two ontologies),
3. alignments with entities involved in more than one correspondence that are denoted by the use of * (zero-or-more) or + (more-than-zero) in their cardinalities.

In case of multiple matching (1), the alignments must contain correspondences relating more than two entities. The definitions above must then be extended accordingly. This is not covered further here.

The second kind of correspondences (2) can be thought of as using non binary relations. However, given the nature of the problem: matching two ontologies, we will consider that these objects can be grouped by operators in the entity language Q_L which can include operators such as concatenation, arithmetic operations or logical connectors for that purpose. In its simplest expression the only construction can be a set.

Option (3) is related to the multiplicity of the alignment if it is considered as a relation. By analogy with mathematical functions, it is useful to define some properties of the alignments. These apply when the only considered relation is equality ($=$) and the confidence measures are not taken into account. One can ask for a total alignment with regard to one ontology, i.e., all the entities of one ontology must be successfully mapped to the other one. This property is purposeful whenever thoroughly transcribing knowledge from one ontology to another is the goal: there is no entity that cannot be translated.

One can also require the alignment to be injective with regard to one ontology, i.e., all the entities of the other ontology is part of at most one correspondence. Injectivity is useful in ensuring that entities that are distinct in one ontology remain distinct in the other one. In particular, this contributes to the reversibility of alignments.

Definition 7 (Total alignment, injective alignment)**.** *Given two ontologies o and o', an alignment A over o and o' is called a* total alignment *from o to o' if and only if:*

$$\forall e \in Q_L(o), \exists e' \in Q_{L'}(o'); \langle e, e', = \rangle \in A$$

and, it is called an injective alignment *from o to o' if and only if:*

$$\forall e' \in Q_{L'}(o'), \exists e_1, e_2 \in Q_L(o); \langle e_1, e', = \rangle \in A \wedge \langle e_2, e', = \rangle \in A \implies e_1 = e_2$$

These properties heavily depend on the ontology entity languages which are chosen for these alignments.

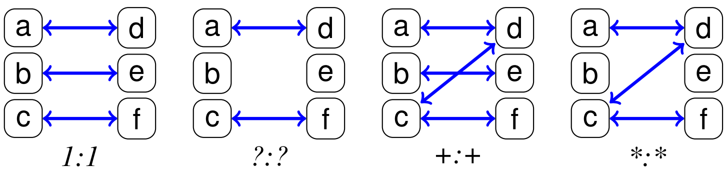
Usual mathematical properties apply to these alignments. In particular, a total alignment from o to o' is a surjective alignment from o' to o. A total alignment from both o and o', which is injective from one of them, is a bijection. In mathematical English, an injective function is said to be *one-to-one* and a surjective function to be *onto*. Due to the wide use among matching practitioners of the term *one-to-one* for a bijective, i.e., both injective and surjective, alignment, we will only use one-to-one for bijective.

Finally, we can extend these definitions when correspondence relations are *not* equivalence. In such a case, they do not ensure the same properties. For instance, injectivity does not guarantee reversibility of the alignment used as a transformation.

In conceptual models and databases, the term multiplicity denotes the constraints on a relation. Usual notations are 1:1 (one-to-one), 1:m (one-to-many), n:1 (many-to-one) or n:m (many-to-many). If we consider only total and injective properties, denoted as 1 for injective and total, ? for injective, + for total and * for none, and the two possible orientations of the alignments, from o to o' and from o' to o, the multiplicities become: ?:?, ?:1, 1:?, 1:1, ?:+, +:?, 1:+, +:1, +:+, ?:*, *:?, 1:*, *:1, +:*, *:+, *:* [6].

Example 5 (Alignment multiplicity)**.** *The alignment of Table 4 is ?:?. If we add the correspondence* Product $\geq$ Volume*, then it is ?:*. If we now consider relating any entity of the second ontology to another entity of the first one, then it becomes ?:+.*

The four pictures below display some of the possible configurations for two ontologies composed of three classes each.



4 Alignment semantics

We provide a simple foundation for alignments because it is useful to know what is expected from a matching algorithm. However, as will be demonstrated, the semantics of alignment provides a definition of how alignments must be interpreted and not of how alignments must be found by a matching algorithm. In this respect, we provide only a semantics for interpreting alignment and not for the matching operation.

The usual way of providing a semantics for related conceptual systems is through modal logic of knowledge and belief [11; 25]. In the line of the work on data integration, we only give a first-order model theoretic semantics. It depends on the semantics of ontologies but does not interfere with it. In fact, given a set of

Figure 3: Equalising semantics.

ontologies and a set of alignments between them, we can evaluate the semantics of the whole system in terms of the semantics of each individual ontology.

The main problem arising is the non compatibility of the domains of interpretation. Given several ontologies, it is possible to consider different positions with regard to the domain of interpretation:

– For all these ontologies, the domain of interpretation D is unique. This approach is useful when ontologies describe a set of well defined entities, like the set of files shared in a peer-to-peer system. This approach has been taken in [4; 3].

– For each ontology o, the domain D_o may be different. Domains are related with the help of domain relations $r_{o,o'}$ which map elements of D_o to corresponding elements of domain $D_{o'}$. This approach is used in [15; 1].

– There is no constraint on the domain of interpretation of ontologies. This is the assumption that will be considered here. For dealing with this assumption, we use a universal domain U, that may be defined as the union of all the domains under consideration, and an equalising function γ or rather a set of equalising functions: $\gamma_o : D_o \longrightarrow U$.

[26] considers the implications of these three models. Here, because the models of various ontologies can have different interpretation domains, we use the notion of an equalising function, which helps make these domains commensurate.

Definition 8 (Equilising function)**.** *Given a family of interpretations $\langle I_o, D_o\rangle_{o\in\Omega}$ of a set of ontologies Ω, an equalising function for $\langle I_o, D_o\rangle_{o\in\Omega}$ is a family of functions $\gamma = (\gamma_o : D_o \longrightarrow U)_{o\in\Omega}$ from the ontology domains of interpretation to a global domain of interpretation U. The set of all equalising functions is called* Γ.

When it is unambiguous, we will use γ as a function. The goal of this γ function is only to be able to (theoretically) compare elements of the domain of interpretation. It is simpler than the use of domain relations in distributed first-order logics [15] in the sense that there is one function per domain instead of relations for each pair of domains.

The equalising functions can be different for each ontology. This means, in particular, that even if two ontologies are interpreted over the same domain of interpretation, it is not compulsory that the equalising function maps their elements to the same element of U, though it remains possible. This allows for a loose coupling of the interpretations.

The relations used in correspondences do not necessarily belong to the ontology languages. As such, they do not have to be interpreted by the ontology semantics. Therefore, we have to provide semantics for them.

Definition 9 (Interpretation of alignment relations)**.** *Given $r \in \Theta$ an alignment relation and U a global domain of interpretation, r is interpreted as a binary relation over U, i.e., $r^U \subseteq U \times U$.*

For the sake of simplicity, we consider correspondences that are only triples of the following form: $\langle e, e', r \rangle$. The definition of correspondence satisfiability relies on γ and the interpretation of relations. It requires that in the equalised models, the correspondences are satisfied.

Definition 10 (Satisfied correspondence)**.** *A correspondence $c = \langle e, e', r \rangle$ is satisfied for an equalising function γ by two models m, m' of o, o' if and only if $\gamma_o \cdot m \in \mathcal{M}(o)$, $\gamma_{o'} \cdot m' \in \mathcal{M}(o')$ and*

$$\langle \gamma_o(m(e)), \gamma_{o'}(m'(e')) \rangle \in r^U$$

This is denoted as $m, m' \models_\gamma c$.

Definition 11 (Satisfied alignment)**.** *An alignment A is satisfied for an equalising function γ by two models m, m' of o, o' if and only if all its correspondences are satisfied for γ by m and m'. This is denoted as $m, m' \models_\gamma A$.*

This is useful for defining the classical notions of validity and satisfiability.

Definition 12 (Alignment validity)**.** *An alignment A of two ontologies o and o' is said to be valid if and only if*

$$\forall m \in \mathcal{M}(o), \forall m' \in \mathcal{M}(o'), \forall \gamma \in \Gamma, m, m' \models_\gamma A$$

This is denoted as $\models A$.

From the practical perspective, this is not a very useful definition since unless the ontologies have no models, it will be very difficult to find valid alignments. A relaxed definition could consider the validity that, given an equalising function, describes how domains are related. Valid alignments are direct logical consequences of the two ontologies. Thus they do not provide additional information than what is already in these ontologies. Satisfiable alignments offer more information.

Definition 13 (Satisfiable alignment)**.** *An alignment A of two ontologies o and o' is said to be satisfiable if and only if*

$$\exists m \in \mathcal{M}(o), \exists m' \in \mathcal{M}(o'), \exists \gamma \in \Gamma; m, m' \models_\gamma A$$

Thus, an alignment is satisfiable if there are models of the ontologies that can be combined in such a way that this alignment makes sense. The satisfiable set of alignments is far larger than the set of valid ones. Again, one can define γ-satisfiable alignments, i.e., alignments satisfiable for a given equalising function.

Given an alignment between two ontologies, the semantics of the aligned ontologies can be defined as follows.

Definition 14 (Models of aligned ontologies)**.** *Given two ontologies o and o' and an alignment A between these ontologies, a model m'' of these ontologies aligned by A is a triple $\langle m, m', \gamma \rangle \in \mathcal{M}(o) \times \mathcal{M}(o') \times \Gamma$, such that $m, m' \models_\gamma A$.*

In that respect, the alignment acts as a model filter for the ontologies. It selects the interpretations of ontologies which are coherent with the alignments. This allows transferring information from one ontology to another since reducing the set of models will entail more consequences in each aligned ontology.

These definitions can be generalised to an arbitrary number of alignments and ontologies captured in the concept of a distributed system [15; 12].

Definition 15 (Distributed system of networked ontologies)**.** *A distributed system of networked ontologies $\langle \Omega, \Lambda \rangle$ is made of a set Ω of ontologies and a set Λ of alignments between these ontologies. We denote as $\Lambda(o, o')$ the set of alignments in Λ between o and o'.*

Definition 16 (Models of distributed systems)**.** *Given a finite set of n ontologies Ω and a finite set of alignments Λ between pairs of ontologies in Ω, a model of the distributed system $\langle \Omega, \Lambda \rangle$ is a $n+1$-uple of models $\langle m_1 \ldots m_n, \gamma \rangle \in \mathcal{M}(o_1) \times \ldots \mathcal{M}(o_n) \times \Gamma$, such that for each alignment $A \in \Lambda(o_i, o_j)$, A is satisfied by $\langle m_i, m_j, \gamma \rangle$.*

This definition coincides with a coherent model of the world in which all models satisfy all alignments. This is the standpoint of an omniscient observer and it corresponds to the global knowledge of a distributed system as defined in [11].

However, if one agent has an inconsistent ontology then the distributed system has no model. Therefore, even agents not connected to the inconsistent ontology cannot compute reasonable models. Moreover, an agent knowing an ontology and the related alignments would like to use the system by gathering information from its neighbours and considering only the models of this information. Thereby, it would be able to compute consequence through some complete deduction mechanisms. This is important when asking agents to answer queries and corresponds to local knowledge in [11]. This is the knowledge an agent can achieve by communicating only with the agents it is connected to in a distributed system.

From that standpoint, there can be several ways to select the acceptable models given the distributed system:

$$\mathcal{M}^1_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \forall A \in \Lambda(o,o'), \exists m' \in \mathcal{M}(o'); m, m' \models_\gamma A\}$$

$$\mathcal{M}^2_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \forall A \in \Lambda(o,o'), \exists m' \in \mathcal{M}^2_{\Omega,\Lambda}(o'); m, m' \models_\gamma A\}$$

$$\mathcal{M}^3_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \exists m' \in \mathcal{M}(o'); \forall A \in \Lambda(o,o'), m, m' \models_\gamma A\}$$

$$\mathcal{M}^4_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \exists m' \in \mathcal{M}^4_{\Omega,\Lambda}(o'); \forall A \in \Lambda(o,o'), m, m' \models_\gamma A\}$$

$$\mathcal{M}^5_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \forall A \in \Lambda(o,o'), \forall m' \in \mathcal{M}^5_{\Omega,\Lambda}(o'); m, m' \models_\gamma A\}$$

$$\mathcal{M}^6_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \forall A \in \Lambda(o,o'), \forall m' \in \mathcal{M}(o'); m, m' \models_\gamma A\}$$

The selection of the γ would ideally be made after the models are chosen because they determine the domains on which the equalising function is built. In practice, the satisfiability condition will only select those equalising functions that are compatible with each of the models.

These approaches have been ordered from the more optimistic to the more cautious. $\mathcal{M}^1_{\Omega,\Lambda}$ selects the models that satisfy each alignment in at least one model of the connected ontology. $\mathcal{M}^3_{\Omega,\Lambda}$ is stronger since it requires that the same model of the connected ontology satisfies all the alignments between the two ontologies. $\mathcal{M}^6_{\Omega,\Lambda}$ is very strong since all alignments must be satisfied by all models of the connected ontologies. $\mathcal{M}^2_{\Omega,\Lambda}$, $\mathcal{M}^4_{\Omega,\Lambda}$ and $\mathcal{M}^5_{\Omega,\Lambda}$ are fixed point characterisations that, instead of considering the initial models of the connected agents, consider their selected models by the same function. This contributes propagating the constraints to the whole connected components of the distributed system. While for $\mathcal{M}^2_{\Omega,\Lambda}$ and $\mathcal{M}^4_{\Omega,\Lambda}$ this strengthens the constraints, for $\mathcal{M}^5_{\Omega,\Lambda}$, this relaxes them with regard to $\mathcal{M}^6_{\Omega,\Lambda}$. Here, an inconsistent model is a problem only to related agents and only for versions $\mathcal{M}^1_{\Omega,\Lambda}$, $\mathcal{M}^3_{\Omega,\Lambda}$ and $\mathcal{M}^4_{\Omega,\Lambda}$ which require the existence of a model for each related ontology.

Each of these options allows the definition of a semantics for distributed systems that is different from the model of distributed system considered above. It is also analogous to the distributed knowledge of the system following [11]. $\mathcal{M}^5_{\Omega,\Lambda}$ would correspond to the semantics given in [12; 2]. A definition of acceptable models for an ontology corresponding to option $\mathcal{M}^4_{\Omega,\Lambda}$ is given as follows.

Definition 17 (Models of an ontology modulo alignments)**.** *Given a distributed system $\langle\Omega, \Lambda\rangle$, the models of $o \in \Omega$ modulo Λ are those models of o, such that for each ontology o' there exists a model that satisfies all elements of Λ between o and*

o':

$$\mathcal{M}^4_{\Omega,\Lambda}(o) = \{m \in \mathcal{M}(o); \exists\gamma \in \Gamma; \forall o' \in \Omega, \exists m' \in \mathcal{M}^4_{\Omega,\Lambda}(o'); \forall A \in \Lambda(o,o'), m, m' \models_\gamma A\}$$

One can be even more restrictive, as in local model semantics [14], by considering only a subset of the possible models of each ontology.

When dealing with ontology matching between a pair of ontologies, the matter of semantics between ontologies is not related to the alignment but to the interpretation of the full distributed system, for instance, depending if one wants to enforce global consistency or not. In this book we will not take a position on such a matter and will only retain the basic interpretation framework provided above.

Note: A distributed system can be seen as an ontology defining a $\models$ relation. It thus can be included within a distributed system as an ontology.

Finally, such a formalism contributes to the definition of the meaning of alignments: it describes what are the consequences of ontologies with alignments, i.e., what can be deduced by an agent. However, it does not describe what the correct alignments are: matching is not a deductive task but an inductive one. The framework is nevertheless particularly useful for deciding if delivered alignments are consistent, i.e., if distributed systems have a model or not. Hence, it is useful for specifying what is expected from matching algorithms and how they should be designed or evaluated.

5 Matching ontologies

The ontology matching problem may be described in one sentence: given two ontologies each describing a set of discrete entities (which can be classes, properties, rules, predicates, or even formulas), find the correspondences, e.g., equivalence or subsumption, holding between these entities. This set of correspondences is called an alignment.

The matching operation determines the alignment A' for a pair of ontologies o and o'. There are some other parameters that can extend the definition of the matching process, namely:

- (i) the use of an input alignment A, which is to be completed by the process;
- (ii) the matching parameters, p, e.g., weights, thresholds; and
- (iii) external resources used by the matching process, r, e.g., common knowledge or domain specific thesauri.

So, the matching process can be seen as a function f which, from a pair of ontologies o and o', an input alignment A, a set of parameters p and a set of resources r, returns an alignment A' between these ontologies:

$$A' = f(o, o', A, p, r)$$

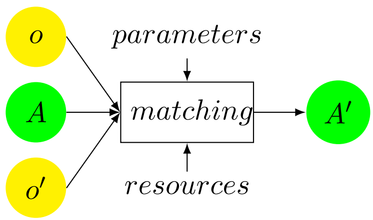


Figure 4: The ontology matching process: it establishes an alignment (A') from two ontologies (o and o') and optionally an input alignment (A), parameters and external resources.

There have already been many reviews of ontology matching algorithms [20; 24; 17; 9], so we will be brief and refer the reader to these presentations.

Ontology matching consists of generating an alignment from two (or more) ontologies. There are many different features of ontologies that are usually used for providing matching:

terminological techniques are based on the text found within ontologies for identifying ontology entities (labels), documenting them (comments) or other surrounding textual sources (related element labels). These techniques come from natural language processing and information retrieval. They can use the string structure themselves, e.g., string distances, the ontology as corpus, e.g., statistical measures based on the frequency of occurrence of a term, or external resources, such as dictionaries.

structural techniques are based on the relations between ontology entities. These can be relations between entities and their attributes, including constraints on their values, or relations with other entities. These techniques take advantage of type comparison techniques or more elaborate graph techniques, e.g., tree distances, path matching, graph matching.

extensional techniques compare the extension of entities. These extensions can be made of other entities, e.g., instances, as well as related resources, e.g., indexed documents. They differ depending on if the two ontologies share resources, e.g., they index the same set of documents, or not (in which case a similarity between the extensions may be established). These techniques can come from data analysis and statistics.

semantic techniques are based on the semantic definition of ontologies. They use extra formalised knowledge and theorem provers for finding consequences of a particular alignment. This can be used for expanding the alignment or, on the contrary, for detecting conflicting correspondences.

Of course, most of the systems combine several techniques in order to improve their results. The techniques can be combined by aggregating distance results [23], by using selection functions for choosing which one to use in the present case [16; 22], or by deeply involving them all in global distance computation [10; 18].

Moreover, there is a difference when training sets are available or not (this is most often useful when a matching algorithm is needed for recognising instances). When available, one can apply machine learning techniques such as Bayes learning, vector support machines or decision trees.

6 Applications of alignment semantics

The goal is to show the incidence of the chosen semantics for particular applications.

6.1 Evaluating alignments

See [8].

6.2 Ontology modules

6.3 Alignment composition

See [26] and manuscript notes.

6.4 Distributed reasoning

7 Conclusion

References

[1] Alexander Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. *Journal on Data Semantics*, I:153–184, 2003.

[2] Paolo Bouquet, Marc Ehrig, Jérôme Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krötzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1, Knowledge web NoE, 2004.

[3] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Logical foundations of peer-to-peer data integration. In *Proc. 23rd Symposium on Principles of Database Systems (PODS)*, pages 241–251, Paris (FR), 2004.

[4] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. A framework for ontology integration. In Isabel Cruz, Stefan Decker, Jérôme Euzenat, and Deborah McGuinness, editors, *The emerging semantic web*, pages 201–214. IOS Press, Amsterdam (NL), 2002.

[5] Pieter De Leenheer and Tom Mens. Ontology evolution; state of the art and future directons. In Martin Hepp, Pieter De Leenheer, Aldo De Moor, and York Sure, editors, *Ontology management: semantic web, semantic web services, and business applications*, chapter 5, pages 131–176. Springer, New-York (NY US), 2008.

[6] Jérôme Euzenat. Towards composing and benchmarking ontology alignments. In *Proc. ISWC Workshop on Semantic Integration*, pages 165–166, Sanibel Island (FL US), 2003.

[7] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture notes in computer science*, pages 698–712, Hiroshima (JP), 2004.

[8] Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 248–253, Hyderabad (IN), 2007.

[9] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2007.

[10] Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proc. 16th European Conference on Artificial Intelligence (ECAI)*, pages 333–337, Valencia (ES), 2004.

[11] Ronald Fagin, Joseph Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge*. The MIT press, Cambridge (MA US), 1995.

[12] Enrico Franconi, Gabriel Kuper, Andrei Lopatenko, and Luciano Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In *Proc. VLDB International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*, pages 64–76, Berlin (DE), 2003.

[13] Avigdor Gal, Ateret Anaby-Tavor, Alberto Trombetta, and Danilo Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *The VLDB Journal*, 14(1):50–67, 2005.

[14] Chiara Ghidini and Fausto Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[15] Chiara Ghidini and Luciano Serafini. Distributed first order logics. In *Proc. 2nd Conference on Frontiers of Combining Systems (FroCoS)*, pages 121–139, Amsterdam (NL), 1998.

[16] Ningsheng Jian, Wei Hu, Gong Cheng, and Yuzhong Qu. Falcon-AO: Aligning ontologies with Falcon. In *Proc. K-CAP Workshop on Integrating Ontologies*, pages 87–93, Banff (CA), 2005.

[17] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[18] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: a versatile graph matching algorithm. In *Proc. 18th International Conference on Data Engineering (ICDE)*, pages 117–128, San Jose (CA US), 2002.

[19] Eric Prud'hommeaux and Andrew Seaborne (ed.). SPARQL query language for RDF. Working draft, W3C, 2007.

[20] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[21] Marie-Christine Rousset, Philippe Adjiman, Philippe Chatalic, François Goasdoué, and Laurent Simon. Somewhere in the semantic web. In *Proc. 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SofSem)*, volume 3831 of *Lecture notes in computer science*, pages 84–99, Merin (CZ), 2006.

[22] Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. Using Bayesian decision for ontology mapping. *Journal of Web Semantics*, 4(1):243–262, 2006.

[23] Willem Robert van Hage, Sophia Katrenko, and Guus Schreiber. A method to combine linguistic ontology-mapping techniques. In *Proc. 4th International Semantic Web Conference (ISWC)*, volume 3729 of *Lecture notes in computer science*, pages 732–744, Galway (IE), 2005.

[24] Holger Wache, Thomas Voegele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. Ontology-based integration of information – a survey of existing approaches. In *Proc. IJCAI Workshop on Ontologies and Information Sharing*, pages 108–117, Seattle (WA US), 2001.

[25] Mike Wooldridge. *Reasoning about rational agents*. The MIT press, Cambridge (MA US), 2000.

[26] Antoine Zimmermann and Jérôme Euzenat. Three semantics for distributed systems and their relations with alignment composition. In *Proc. 5th International Semantic Web Conference (ISWC)*, volume 4273 of *Lecture notes in computer science*, pages 16–29, Athens (GA US), 2006.