OpenKnowledge

FP6-027253

Ontology Handling Framework

Marco Schorlemmer[1]
with input from
Manuel Atencia[1], Alan Bundy[2], Fausto Giunchiglia[3],
Vanessa Lopez[4], and Fiona McNeill[2]

[1] Artificial Intelligence Research Institute, IIIA-CSIC, Spain
[2] School of Informatics, University of Edinburgh, UK
[3] Dept of Information and Communication Technology, University of Trento, Italy
[4] Knowledge Media Institute, The Open University, UK

Report Version: final
Report Preparation Date:
Classification: deliverable D3.2
Contract Start Date: 1.1.2006 Duration: 36 months
Project Co-ordinator: University of Edinburgh (David Robertson)
Partners: IIIA(CSIC) Barcelona
Vrije Universiteit Amsterdam
University of Edinburgh
KMI, Open University
University of Southampton
University of Trento

OpenKnowledge Deliverable D3.2

A Formal Conceptual Framework for Semantic Matching, Alignment, and Refinement in P2P Information Systems*

Marco Schorlemmer[1]

with input from
Manuel Atencia,[1] Alan Bundy,[2] Fausto Giunchiglia,[3]
Vanessa López,[4] and Fiona McNeill[2]

[1]IIIA - Artificial Intelligence Research Institute, CSIC, Spain
[2]School of Informatics, The University of Edinburgh, UK
[3]Dept. of Information and Communication Technology, University of Trento, Italy
[4]Knowledge Media Insitute, Open University, UK

Abstract

We specify a formal conceptual framework in which to characterise semantic matching and alignment. Our aim with this framework is threefold: (a) to provide concrete definitions of the concepts at work; (b) to describe, in a unifying manner, various different ontology matching approaches —at least those feeding into the OpenKnowledge project; and (c) to be useful for pinning down several formal issues arising in semantic matching and alignment in the context of P2P information systems, such as composition of semantic alignments, dynamic ontology refinement, and semantic-alignment interaction models.

1 Introduction

In order for two systems (databases, agents, peers, components, etc.) to be considered semantically integrated, both will have need to commit to a shared conceptualisation of the application domain. Commonly, this is achieved by providing an explicit specification of this conceptualisation —what has become to be known as an *ontology*— and by defining each system's local vocabulary in terms of the ontology's vocabulary. This sort of integration is dubbed "semantic" precisely because it assumes that the ontology is some sort of structured

*Original title as stated in the contract with the European Commission: "Specification of a common framework for characterising contextual ontology and context mapping."

theory T —coming thus equipped with a precise semantics for the structure it holds— and because each system's local language L_i is *interpreted* in T (e.g., in the technical sense of a theory interpretation as defined in [End02], when T is a theory in first-order logic). Semantic integration is therefore always relative to the theory T into which local languages are interpreted. We shall call this theory the *reference theory* of the integration.

The use of ontologies as reference theories for semantic integration, however, is more in tune with a classical codification-centered knowledge management tradition, as put forward by Corrêa da Silva and Agustí [CA03]. Such tradition comprises the efforts to define standard upper-level ontologies such as CyC [Len95] and SUO[1], or to establish public ontology repositories for specific domains to favour knowledge reuse such as the Ontolingua server [FFR97]. Corrêa da Silva and Agustí remark that "centralised ontologies [...] promise to bring the control of the organisation back to what was possible under classical management techniques. The problem is that they may also bring back the rigidity of agencies organised under the classical management tenets."

Since ontologies are the result of an inter-subjective agreement among individuals about the same fragment of the objective world, they are also highly context-dependent and hardly will result to be general-purpose, regardless of how abstract and upper-level they might be. This is even more true in highly distributed, open, and dynamic environments such as P2P information systems. In these sort of environments it is more realistic to achieve certain levels of semantic integration by matching vocabulary on the fly. This actually means that semantic integration has to occur with respect to a theory that is not explicitly given *a priori*, but which one could infer from the matching process, nevertheless. We shall call it the *virtual* reference theory.

In this deliverable we specify a formal conceptual framework in which to characterise semantic matching and alignment. Our aim with this framework is threefold: (a) to provide concrete definitions of the concepts at work (Section 2); (b) to describe, in a unifying manner, various different ontology matching approaches —at least those feeding into the OpenKnowledge project (Section 3); and (c) to be useful for pinning down several formal issues arising in semantic matching and alignment in the context of P2P information systems, such as composition of semantic alignments, dynamic ontology refinement, and semantic-alignment interaction models (Section 4).

2 The Framework: Basic Concepts and Definitions

We shall be concerned with semantic integration understood as the integration of two systems by virtue of the interpretation of their respective vocabularies into a reference theory, expressible in some logical language.

By *vocabulary* we mean a set V of words and symbols used by a system to represent and organise its local knowledge. In a formal, logic-based representation language the vocabulary is constituted by the non-logical symbols used to form sentences and formulas (in this case it is usually referred to as *parameters* or *signature*). The *language* is then the set $L(V)$ of all well-formed formulae over

[1] `http://suo.ieee.org`

a given vocabulary V. We shall also write L when we do not want to explicitly refer to the vocabulary. We call the elements of a language L, *sentences*.

In declarative representation languages, knowledge is represented and organised by means of theories. DL-based ontologies are such an example. A convenient way to abstractly characterise theories in general is by means of the notion of consequence relation. Given a language L, a *consequence relation* over L is, in general, a binary relation $\vdash$ on subsets of L which satisfying certain structural properties.[2] Consequence relations are also suitable to capture other sorts of mathematical structures used to organise knowledge in a systematic way, such as taxonomic hierarchies. When defined as a binary relation on L (and not on subsets of L), for instance, it coincides with a partial order. Furthermore, there exists a close relationship between consequence relations and classification relations (which play a central role in ontological knowledge organisation), which has been thoroughly studied from a mathematical perspective in [DH01, BS97, GW99].

We call a *theory* a tuple $T = \langle L_T, \vdash_T \rangle$, where $\vdash_T \subseteq \mathcal{P}(L_T) \times \mathcal{P}(L_T)$ is a consequence relation, hence capturing with this notion the formal structure of an ontology in general. Finally, in order to capture the relationship between theories, we call a *theory interpretation* a map between the underlying languages of theories that respects consequence relations. That is, a function $i : L_T \to L_{T'}$ is a theory interpretation between theories $T = \langle L_T, \vdash_T \rangle$ and $T' = \langle L_{T'}, \vdash_{T'} \rangle$ if, and only if, for all $\Gamma, \Delta \subseteq L$ we have that $\Gamma \vdash_T \Delta$ implies $i(\Gamma) \vdash_{T'} i(\Delta)$ (where $i(\Gamma)$ and $i(\Delta)$ are the set of direct images of Γ and Δ along i, respectively).[3]

2.1 Semantic Matching

We call *semantic matching* to the process that takes two theories T_1 and T_2 as input (called *local theories*) and computes a third theory $T_{1 \leftrightarrow 2}$ as output (called *bridge theory*) that captures the semantic alignment of T_1 and T_2's languages, and which underlies the semantic integration of T_1 and T_2 with respect to a reference theory T. It is important to make a couple of remarks here.

First, one usually distinguishes a theory from its presentation. If the language L is infinite (as for instance in propositional or first-order languages, where the set of well-formed formulae is infinite, despite of having a finite vocabulary), any consequence relations over L will be infinite as well. Therefore, one deals in practice with a finite subset of $\mathcal{P}(L) \times \mathcal{P}(L)$, called a *presentation*, to stand for the smallest consequence relation containing this subset.

A presentation may be empty, in which case the smallest consequence relation over a language L containing it is called the *trivial theory*. We will write $Tr(L)$ for the trivial theory over L. It is easy to proof that, for all $\Gamma, \Delta \subseteq L$, $\Gamma \vdash_{Tr(L)} \Delta$ if, and only if, $\Gamma \cap \Delta \neq \emptyset$.

Rigorously speaking, a semantic matching process actually takes two presentations of local theories as input and computes a presentation of the bridge theory as output. But, from a conceptual perspective, we shall characterise semantic matching always in terms of the theories themselves.

[2] These are commonly those of Identity, Weakening and Global Cut (see [DH01, BS97]).

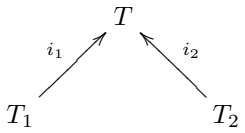
[3] Theories and theory interpretations as treated here can also be seen as particular cases of the more general framework provided by institution theory, which has been thoroughly studied in the field of algebraic software specification (see [GB92]).

Second, the reference theory T is usually *not* an explicit input of the semantic matching process (not even a presentation of it). Instead it should be understood as the background knowledge used by a semantic matcher to infer semantic relationships between the underlying vocabularies of the respective input theories. For a manual matcher, for instance, the reference theory may be entirely dependent on user input, while a fully automatic matcher would need to rely on automatic services (either internal or external to the matcher) to infer such reference theory. It is for this reason that we talk of a *virtual* reference theory, since it is not explicitly provided to the semantic matcher, but is implicit in the way external and internal sources are brought into the matching process as background theory for semantic matching.

Next, we provide precise definitions of what we mean for a bridge theory to capture a semantic alignment of languages, and also what we mean by a semantic alignment underlying a semantic integration of local theories.

2.2 Integration Theory

Definition 1 Two theories T_1 and T_2 are *semantically integrated with respect to* T, if there exist theory interpretations $i_1 : T_1 \to T$ and $i_2 : T_2 \to T$.



We call $\mathcal{I} = \{i_i : T_i \to T\}_{i=1,2}$ the *semantic integration* of *local theories* T_1 and T_2 with respect to *reference theory* T. Two languages L_1 and L_2 are *semantically integrated with respect to* T if their respective trivial theories are.

In semantic matching we are interested in determining the semantic relationship between the languages L_{T_1} and L_{T_2} on which semantically integrated theories T_1 and T_2 are expressed. Therefore, a semantic integration $\mathcal{I}$ of T_1 and T_2 with respect to a reference theory T as defined above is not of direct use, yet. What we would like to have is a theory $T_{\mathcal{I}}$ over the combined language $L_{T_1} \uplus L_{T_2}$ (the disjoint union) expressing the semantic relationship that arises by interpreting local theories in T. We call this the *integration theory* of $\mathcal{I}$, and it is defined as the inverse image of the reference theory T under the sum of the theory interpretations in $\mathcal{I}$. Following are the precise definitions.

Definition 2 Let $i : T \to T'$ be a theory interpretation. The *inverse image* of T' under i, denoted $i^{-1}[T']$, is the theory over the language of T such that $\Gamma \vdash_{i^{-1}[T']} \Delta$ if, and only if, $i(\Gamma) \vdash_{T'} i(\Delta)$.

It is easy to proof that, for every theory interpretation $i : T \to T'$, T is a *subtheory* of $i^{-1}[T']$, i.e., $\vdash_T \subseteq \vdash_{i^{-1}[T']}$.

Definition 3 Given theories $T_1 = \langle L_{T_1}, \vdash_{T_1} \rangle$ and $T_2 = \langle L_{T_2}, \vdash_{T_2} \rangle$, the *sum* $T_1 + T_2$ *of theories* is the theory over the sum of language (i.e., the disjoint union of languages) $L_{T_1} \uplus L_{T_2}$ such that $\vdash_{T_1+T_2}$ is the smallest consequence relation such that $\vdash_{T_1} \subseteq \vdash_{T_1+T_2}$ and $\vdash_{T_1} \subseteq \vdash_{T_1+T_2}$.

Given theory interpretations $i_1 : T_1 \to T$ and $i_2 : T_2 \to T$, the *sum* $i_1 + i_2 : T_1 + T_2 \to T$ *of theory interpretations* is just the sum of their underlying map of languages.

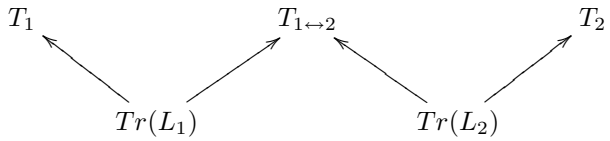
Definition 4 Let $\mathcal{I} = \{i_{1,2} : T_{1,2} \to T\}$ be a semantic integration of T_1 and T_2 with respect to T. The *integration theory* $T_{\mathcal{I}}$ of the semantic integration $\mathcal{I}$ is the inverse image of T under the sum of interpretations $i_1 + i_2$, i.e. $T_{\mathcal{I}} = (i_1 + i_2)^{-1}[T]$.

The integration theory faithfully captures the semantic relationships between sentences in L_{T_1} and L_{T_2} as determined by their respective interpretation into T, but expressed as a theory over the combined language $L_{T_1} \uplus L_{T_2}$. The sum of local theories $T_1 + T_2$ is therefore always a subtheory of the integration theory $T_{\mathcal{I}}$, because it is through the interpretations in T where we get the semantic relationship between languages. It captures and formalises the intuitive idea that an integration is more than just the sum of its parts.

2.3 Semantic Alignment

In semantic matching one usually isolates as output to the matching process the bit that makes $T_{\mathcal{I}}$ genuinely a supertheory of $T_1 + T_2$. The idea is to characterise a theory $T_{1\leftrightarrow 2}$ over a the disjoint union of subsets $L_1 \subseteq L_{T_1}$ and $L_2 \subseteq L_{T_2}$, called *bridge theory*, which, together with T_1 and T_2, uniquely determines the integration theory $T_{\mathcal{I}}$. To keep everything characterised uniformly in the same conceptual framework, the bridge theory, together with its relationship to the local theories T_1 and T_2, can be expressed by a diagram of theory interpretations as follows.

Definition 5 A *semantic alignment* $\mathcal{A}$ *of* T_1 *with* T_2 is a diagram



in the category of theories and theory interpretations, where $L_i \subseteq L_{T_i}$ and $T_{1\leftrightarrow 2}$ is a theory whose underlying language $L_{T_{1\leftrightarrow 2}} = L_1 \uplus L_2$, and where all arrows are theory inclusions. We shall also write $T_1 \overset{\mathcal{A}}{\longleftrightarrow} T_2$ as a shorthand of an alignment.

We say that a semantic alignment $\mathcal{A}$ *underlies a semantic integration* $\mathcal{I}$ when the colimit of $\mathcal{A}$ in the category of theories and theory interpretation (which always exists) is the integration theory of $\mathcal{I}$, i.e., $colim(\mathcal{A}) = T_{\mathcal{I}}$.

Open Problem: Given a semantic integration $\mathcal{I}$, which are the smallest subsets L_1 and L_2, and which is the smallest bridge theory $T_{1\leftrightarrow 2}$ with $L_1 \uplus L_2$ as its underlying language forming a semantic alignment of T_1 with T_2 such that it underlies $\mathcal{I}$?

2.4 Related Work

The original effort to develop an formal approach to ontology alignment using theory and theory interpretation around the issues of organising and relating ontologies is Kent's Information Flow Framework (IFF) [Ken00]. Recently, Kent has proposed a formal characterisation of semantic integration in terms of IFF [Ken05]. Also recently, Goguen has shown that Kent's approach can be expressed in terms of institution theory [GB92], and he uses this insight to provide foundations for principled semantic integration [Gog05].

The representation of an ontology alignment as a system of objects and morphisms in a category, and of semantic integration by means of a colimit of such a diagram, bears a close relationship to the notion of *W-alignment diagram* described in [ZKEH06]. This is so because both notions share the same categorical approach to semantic alignment. But, unlike in [ZKEH06], we specifically further take the a dual "type-token" structure of semantic integration into account, and we define alignment with respect to this two-tier model. We claim that in this way we better capture Barwise and Seligman's basic insight that that "information flow involves both types and their particulars" [BS97]. This will become clearer when discussing the following examples.

3 Examples of Semantic Matching and Alignment

We have outlined a formal framework that characterises semantic matching in an abstract way, namely as the process that, given two local theories T_1 and T_2 computes a bridge theory $T_{1\leftrightarrow 2}$ that captures the semantic alignment of T_1 and T_2 underlying a semantic integration $\mathcal{I}$ of T_1 and T_2. According to this framework, what conceptually distinguishes particular instances of semantic matching is the integration $\mathcal{I}$, which is basically (a) the virtual reference theory T with respect to which semantic integration occurs, and (b) the theory interpretations i_1 and i_2 capturing the way local vocabularies are interpreted in this reference theory. Let us now illustrate these conceptual differences by describing several semantic matching systems feeding into OpenKnowledge [AKtv06, GSY05, SK05] as particular instances of our general conceptual framework outlined so far.

3.1 Matching Vocabularies by Anchoring to Web Ontologies

Input to the matching process described in [AKtv06] are two sets of terms V_1 and V_2, i.e., two semantically poor ontologies consisting only of unstructured (flat) lists. In order to find semantic matches between elements of V_1 with elements of V_2, the matching process attempts first to map elements of each V_i to concepts occurring in a common background-knowledge ontology O as one may found available on the web, and then use the rich semantic structure of such ontology to infer the semantic relationship between elements of the vocabulary. The first step of this process in called *anchoring*, and is typically done in a computationally cheap fashion doing lexical matching (although in some cases it is supplemented with expert assistance). During the anchoring step not all

elements of V_i are usually mapped, and it yields n-m relationships between elements of V_i and concepts in O.

In the second step of the process, namely when reasoning with the semantically rich ontology O, one now is able to infer semantic relations such as subsumption, equivalence, or overlap between elements of the initially unstructured vocabularies by virtue of them being anchored to concepts in O, which yields much more matches than using lexical matching only. Thus, if a term $s_1 \in V_1$ is mapped to a concept c in O (e.g. "Aorta thorcalis dissection" is anchored to "Aorta") which happens to be a subconcept of a concept d (e.g., "Artery") to which a term $s_2 \in V_2$ is anchored (e.g., "Dissection of artery"), then a relationship of the kind "is more specific than" can be inferred between s_1 ("Aorta thorcalis dissection") and s_2 ("Dissection of artery").

Although [AKtv06] do not give precise descriptions of which kind of semantic relationships are inferred, nor how they are inferred from the background knowledge ontology, the process described above can still be seen as a particular kind of semantic integration as formalised in Section 2. For that reason we shall conceptualise source and target vocabularies and the background ontologies, as well as anchoring, in terms of theories and their interpretations.

Take the local theories $Tr(V_1')$ and $Tr(V_2')$ to be the trivial theories over the sub-vocabularies $V_1' \subseteq V_1$ and $V_2' \subseteq V_2$ for which each element can be anchored to a background ontology O. Notice that here languages and vocabularies as defined in Section 2 coincide. For the reference theory the obvious theory would be to take the theory T_O determined by O (see [KS03]). If O is just a taxonomy, for instance, each sub-/super concept relation $c \leq d$ in O can be captured by $\{c\} \vdash \{d\}$ being a constraint in T_O. Anchoring, however, may map a local term in V_i' to more than one concept in O, which makes it impossible to formalise anchoring as theory interpretations from local theories T_i to T_O. For that reason we shall take the *disjunctive power* of T_O to be the reference theory $T = \vee T_O$, defined as follows: Let $T_O = \langle L_{T_O}, \vdash_{T_O} \rangle$; $\vee T_O = \langle L_{\vee T_O}, \vdash_{\vee T_O} \rangle$ where $L_{\vee T_O} = \mathcal{P}(L_{T_O})$, the powerset of L_{T_O}, and for each $\Gamma, \Delta \subseteq L_{\vee T_O}$, $\Gamma \vdash_{\vee T_O} \Delta$ if and only if, for all $Y \subseteq L_{T_O}$ such that $Y \cap X \neq \emptyset$ for each $X \in \Gamma$, we have that $Y \vdash_{T_O} \bigcup \Delta$.

The following proposition states that anchoring, i.e., mapping elements from each V_i to subsets of concepts of O are indeed theory interpretations.

Proposition 1 *Anchoring are theory interpretations.*

PROOF: Since anchoring is applied to a semantically poor ontology, namely just a subset V_i' of terms, which we have characterised by its trivial theory $Tr(V_i')$, any function on V will trivially be a theory interpretation: Let $a_i : V_i' \rightarrow \mathcal{P}(L_{T_O})$ be the anchoring function. $\Gamma \vdash_{Tr(V_i')} \Delta$ iff $\Gamma \cap \Delta \neq \emptyset$. Consequently, $a_i(\Gamma) \cap a_i(\Delta) \neq \emptyset$, and therefore $a_i(\Gamma) \vdash_{\vee T_O} a_i(\Delta)$. □

A direct consequence of the above proposition is the following corollary:

Corollary 1 $\mathcal{I} = \{a_i : Tr(V_i') \rightarrow \vee T_O\}_{i=1,2}$ *is a semantic integration*

The matching process described in [AKtv06] may indeed be seen in terms of the integration theory of $\mathcal{I}$. Actually, the formalisation of anchoring as a semantic integration $\mathcal{I}$ allows us to precisely define the semantic relationships inferred by the process that constitute the integration theory of $\mathcal{I}$.

Definition 6 Let $\mathcal{I} = \{a_i : Tr(V_i') \rightarrow \vee T_O\}_{i=1,2}$ be the the semantic integration defined above.

- We say that $s_1 \in V_1$ *is more specific than* $s_2 \in V_2$ iff $\{s_1\} \vdash_{T_\mathcal{I}} \{s_2\}$. Consequently, s_1 is more specific than s_2 iff for each concept $c \in O$ to which s_1 is anchored there exists a concept $d \in O$ to which s_2 is anchored such that, $c \leq d$ in O.
- We say that $s_1 \in V_1$ *is more general than* $s_2 \in V_2$ iff $\{s_2\} \vdash_{T_\mathcal{I}} \{s_1\}$. Consequently, s_1 is more general than s_2 iff for each concept $d \in O$ to which s_2 is anchored there exists a concept $c \in O$ to which s_1 is anchored such that, $d \leq c$ in O.
- We say that $s_1 \in V_1$ *is equivalent to* $s_2 \in V_2$ iff $\{s_1\} \vdash_{T_\mathcal{I}} \{s_2\}$ and $\{s_2\} \vdash_{T_\mathcal{I}} \{s_1\}$. Consequently, s_1 is equivalent to s_2 iff for each concept $c \in O$ to which s_1 and for each concept $d \in O$ to which s_2 is anchored such that, $c \equiv d$ in O.
- We say that $s_1 \in V_1$ *and* $s_2 \in V_2$ *overlap* iff $\{s_1, s_2\} \nvdash_{T_\mathcal{I}} \emptyset$. Consequently, s_1 and s_2 overlap iff there exist a concept $c \in O$ to which s_1 is anchored and a concept $d \in O$ to which s_2 is anchored such that, c and d overlap in O.

3.2 Semantic Matching with S-Match

Input to S-Match [GSY05] are two labelled directed acyclic graphs $G_1 = (N_1, E_1, l_1)$ and $G_2 = (N_2, E_2, l_2)$ with sets of nodes N_1 and N_2, sets of edges $E_1 \subseteq N_1 \times N_1$ and $E_2 \subseteq N_2 \times N_2$, and labelling functions $l_1 : N_1 \rightarrow S_1$ and $l_2 : N_2 \rightarrow S_2$, respectively, where S_1 and S_2 are sets of labels. These input graphs are to be understood as concept taxonomies, such that, given an edge (n, m), $l_i(n)$ denotes a subconcept of $l_i(m)$. S-Match maps nodes of each graph $G_i = (N_i, E_i, l_i)$, with $i = 1, 2$, to formulae in Propositional Description Logic whose atomic concepts are WordNet senses. First, each label $s \in S_i$ in graph G_i is mapped to a formula $f_i(s)$ in Propositional Description Logic (called *concept of label*); second, each node $n \in N_i$ is mapped to a formula $i_i(n)$ in Propositional Description Logic (called *concept at node*) defined as follows:

$$i_i(n) = \bigsqcap_{m \in \uparrow n} f_i(l_i(m))$$

where $\uparrow n$ denotes the set of all nodes reachable from n (including itself). Finally, formulas in Propositional Description Logic can be converted into an equivalent formula in a propositional logic language with Boolean semantics.

Central to the way S-Match computes the semantic relationships between nodes in N_1 and nodes in N_2 is the background knowledge brought into the matcher. S-Match uses for this purpose a library of so called element-level matchers (ranging from string-based matchers looking for shared prefixes, suffixes, computing edit distances and the like, to sense-based matchers such as those based on WordNet's hyper-/hyponym and holo-/meronym structures), which determine a set K of semantic relationships $s\ R\ t$ between labels, with $s \in S_1$, $t \in S_2$, and $R \in \{\sqsubseteq, \sqsupseteq, =, \perp\}$. S-Match's final output is a collection of semantic relationships $n\ R\ m$ between nodes (called *mapping elements*), with $n \in N_1$, $m \in N_2$, and $R \in \{\sqsubseteq, \sqsupseteq, =, \perp\}$, such that

- $n \sqsubseteq m$ iff A_K implies $i_1(n) \rightarrow i_2(m)$ in propositional logic
- $n \sqsupseteq m$ iff A_K implies $i_2(m) \rightarrow i_1(n)$ in propositional logic
- $n = m$ iff A_K implies $i_1(n) \leftrightarrow i_2(m)$ in propositional logic
- $n \perp m$ iff A_K implies $\neg(i_1(n) \wedge i_2(m))$ in propositional logic

where A_K is the set of propositional axioms determined by the background knowledge K as follows:

- $f_1(s) \rightarrow f_2(t) \in A_K$ iff $s \sqsubseteq t$
- $f_2(s) \rightarrow f_1(t) \in A_K$ iff $s \sqsupseteq t$
- $f_1(s) \leftrightarrow f_2(t) \in A_K$ iff $s = t$
- $\neg(f_1(s) \wedge f_2(t)) \in A_K$ iff $s \perp t$

The semantic relationships are computed and checked using a SAT prover. S-match is run for any pair of nodes. Thus, to match two graphs of n and m nodes respectively, S-match has to be executed $n \times m$ times, once for each pair of nodes.

We show that S-Match is indeed a particular instance of our general semantic matching framework. Conceptually, the axioms A_K of the background knowledge fed into S-Match's SAT prover determine a consequence relation $\vdash_{A_K}$ over the propositional language $L_{WordNet}$ whose atomic propositions are WordNet senses: let $\vdash_{A_K}$ be the smallest Boolean consequence relation[4] such that $\emptyset \vdash_{A_K} \{\varphi\}$ if, and only if, $\varphi \in A_K$. Nodes of input graphs are mapped into sentences of $L_{WordNet}$, such that the output semantic relationship between two nodes is determined by the way $\vdash_{A_K}$ relates the sentences into which these nodes are mapped. Consequently, an application of S-Match, with its library of element-level matchers, determines a virtual reference theory $T = \langle L_{WordNet}, \vdash_{A_K} \rangle$.

In addition, an application of S-Match also determines maps of languages $i_i : N_i \rightarrow L_T$. The way these maps are computed make them actual theory interpretations, because S-Match takes the structure of the input graphs into account for the computation of i_i. Since graphs $G_i = (N_i, E_i, l_i)$ are understood as classification taxonomies, they can be characterised as theories $T_i = \langle N_i, \vdash_{E_i} \rangle$, where $\vdash_{E_i}$ are the smallest consequence relations that include E_i. That is, we take nodes as sentences of their languages, and edges as presentations of their theories. The following proposition states that i_i are indeed a theory interpretations:

Proposition 2 *Let $G_i = (N_i, E_i, l_i)$ be a labelled directed acyclic graph, with $l_i : N_i \rightarrow S_i$; let $f : S_i \rightarrow L_T$ be a function mapping labels to propositional formulae in L_T; let $T_i = \langle N_i, \vdash_{E_i} \rangle$ be the theory where $\vdash_{E_i}$ is the smallest consequence relation including E_i; let T be a Boolean theory over L_T.*

The map $i_i : N_i \rightarrow L_T$ defined for all $n \in N_i$ as

$$i_i(n) = \bigwedge_{m \in \uparrow n} f_i(l_i(m))$$

[4] A Boolean consequence relation is a consequence relation that takes the Boolean structure of sentences into account: For instance, if $\vdash$ is Boolean, then $s, t \vdash u$ and $s, t \vdash v$ imply $s \vdash (t \rightarrow u \wedge v)$.

is a theory interpretation $i_i : T_i \rightarrow T$.

PROOF: Suppose $\Gamma \vdash_{T_i} \Delta$. It follows from the definition of $\vdash_{T_i}$ that there exist $n \in \Gamma$ and $m \in \Delta$ such that m is reachable from n in G_i, i.e., $m \in\uparrow n$. Consequently, by the way i_i is defined, $i_i(m)$ is a conjunct of $i_i(n)$. Therefore, $\{i_i(n)\} \vdash_T \{i_i(m)\}$, and consequently $i_i(\Gamma) \vdash_T i_i(\Delta)$. □

A direct consequence of the above proposition is the following corollary:

Corollary 2 $\mathcal{I} = \{i_i : T_i \rightarrow T\}_{i=1,2}$ *is a semantic integration*

Finally we need to proof that S-Match's output indeed captures the integration theory of $\mathcal{I}$:

Theorem 1 *For all* $n \in N_1$ *and for all* $m \in N_2$,

- $n \sqsubseteq m$ *iff* $\{n\} \vdash_{T_\mathcal{I}} \{m\}$
- $n \sqsupseteq m$ *iff* $\{m\} \vdash_{T_\mathcal{I}} \{n\}$
- $n = m$ *iff* $\{n\} \vdash_{T_\mathcal{I}} \{m\}$ *and* $\{m\} \vdash_{T_\mathcal{I}} \{n\}$
- $n \perp m$ *iff* $\{n, m\} \vdash_{T_\mathcal{I}} \emptyset$

PROOF: $n \sqsubseteq m$ is a mapping element iff A_K implies $i_1(n) \rightarrow i_2(m)$, where A_K is the set of propositional formulae axiomatising the background knowledge K determined by S-Match's element-level matchers. By the definition of Boolean consequence relation $\vdash_{A_K}$ of the integration's reference theory, the above implication is equivalent to state that $\{i_1(n)\} \vdash_{A_K} \{i_2(m)\}$. Finally, $\{i_1(n)\} \vdash_{A_K} \{i_2(m)\}$ iff $\{n\} \vdash_{T_\mathcal{I}} \{m\}$. The remaining equivalences are proved analogously. □

3.3 Information-Flow-based Semantic Matching

Input to IF-Map [SK05] are two so called *populated ontologies*, which are partially ordered sets $O_1 = (N_1, \leq)$ and $O_2 = (N_2, \leq)$ of concept names together sets X_1 and X_2 of instances and classification relations $C_1 \subseteq X_1 \times N_1$ and $C_2 \subseteq X_2 \times N_2$ of instances to concept names. IF-Map's output is a consequence relation over the disjoint union $N_1 \uplus N_2$ of concept names capturing the semantic relationship between concepts of one ontology with concepts of the other.

Central to the way semantic relationships are computed is the assumption that an agent gets to know about the conceptualisation of another agent by exchanging instances classified under the concept names of their respective local populated ontology. Consequently, IF-Map's semantic integration will be partial, relative to the set of exchanged instances, and hence will be determined both at the concept and at the instance level:

- An agent i will have attempted to explain a subset N_i' of its concept names to other agents, and
- another agent j will have exchanged with it a subset $Y_j \subseteq X_j$ of its instances.

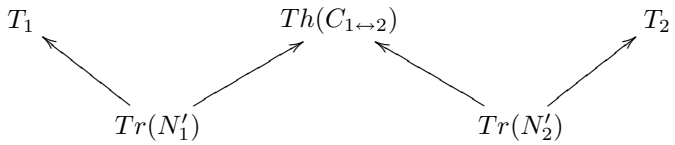
Central for semantic integration to occur is that agent i is capable of computing a classification $C_i' \subseteq N_i' \times X_i'$, where $X_i' = X_i \cup Y_j$, as this allows one to define the classification $C_{1\leftrightarrow 2} \subseteq (Y_1 \cup Y_2) \times (N_1' \uplus N_2')$ of exchanged instances to communicated concept names, which will determine the semantic integration of the agent's ontology theories. This semantic integration of sublanguages N_1' and N_2' is based on the fact that a classification C always determines a theory $Th(C)$ on the language of its concept names (see [BS97] for details).

As in the proof of Proposition 1, it is trivial to see that the inclusion functions i_i sending N_i' into the disjoint union $N_1' \uplus N_2'$ are theory interpretations from $Tr(N_i')$ to $Th(C_{1\leftrightarrow 2})$. Consequently we have:

Proposition 3 *$\mathcal{I} = \{i_i : Tr(N_i') \rightarrow Th(C_{1\leftrightarrow 2})\}$ is a semantic integration.*

Corollary 3 *The reference theory of $\mathcal{I}$ is also its integration theory, i.e., $T_{\mathcal{I}} = Th(C_{1\leftrightarrow 2})$.*

A partial order of concept names $O_i = (N_i, \leq)$ can be characterised as a theory $T_i = \langle N_i, \vdash_{O_i} \rangle$, where $\vdash_{O_i}$ is the smallest consequence relation that includes $\leq$. Consequently, The partial semantic integration $\mathcal{I}$ of subsets N_1' and N_2', determines a semantic alignment of local theories T_1 and T_2 underlying the input ontologies to IF-Map.



The colimit of this alignment determines a semantic integration of the ontologies that is the result of the partial semantic integration achieved by means of communicating concept names and exchanging instance and classification information.

3.4 Interpretation- vs. Classification-based Semantic Matching

The above description of S-Match and IF-Map with respect to the conceptual framework we have outlined illustrates that both are semantic matching techniques that follow two alternative and complementary approaches to semantic integration.

S-Match follows what we call an *interpretation-based* paradigm. Semantic integration is achieved by virtue of how local terminology is interpreted in a virtual reference theory that is implicit to how the matcher uses external or internal sources. In S-Match it is a theory determined by the axioms in a propositional logic yielded by the consultation of a library of element-level matchers. The local context is respected by the way the interpretation is done, namely by respecting the local structure, captured as local theory. In this paradigm it is assumed that the matching process is capable of computing this interpretations in a reference theory.

IF-Map, on the contrary, follows a *classification-based* paradigm. Meaning of local terminology is determined not by way of theory interpretations, but by

how instances are locally classified to sentences of the local language. The virtual reference theory is determined by the semantic alignment that arises from the partial semantic integration through a theory generated by the shared classification system. In this paradigm it is assumed that local agents are capable of classifying any new instance coming from foreign agents according to its own terminology, and vice versa. Other classification-based semantic matchers are [WG02, vBD$^+$05].

4 Semantic Reconciliation in P2P Systems

In this section we explore three issues arising in semantic matching and alignment in open P2P systems, and attempt to put them in relationship with the formal framework outlined above.

4.1 Composing Alignments

Central in query forwarding in a P2P system is the issue of successive translations of a query along different paths in a P2P network. Our aim in this subsection is to formalise the idea of composition of semantic alignments in order to pin down the problem that arises with successive query translation. For this reason we first define the concept of translation with respect to a semantic alignment. It follows the notion of translation of sentences in declarative languages as formalised in [CN00]. In the following, let T_i be theories with underlying languages L_i, and let s_i be a sentence of language L_i.

Definition 7 We say that $s_2 \in L_2$ is a *partial translation* of $s_1 \in L_2$ with respect to semantic alignment $\mathcal{A}$ when $\{s_1\} \vdash_{colim(\mathcal{A})} \{s_2\}$.

We say that $s_2 \in L_2$ is a *translation* of $s_1 \in L_2$ with respect to semantic alignment $\mathcal{A}$ when $\{s_1\} \dashv\vdash_{colim(\mathcal{A})} \{s_2\}$.

We now use the above notion of translation to provide a definition of the composition of semantic alignments.

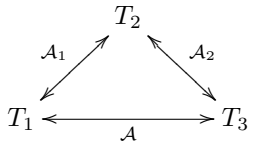
Definition 8 We say that an alignment $\mathcal{A}$ of T_1 with T_3 is a composition of the alignment $\mathcal{A}_1$ of T_1 with T_2 and the alignment $\mathcal{A}_2$ of T_2 with T_3 when the following propositions are equivalent:

1. $s_3 \in L_3$ is a partial translation of $s_1 \in L_1$ with respect to $\mathcal{A}$
2. there exists $s_2 \in L_2$ such that $s_3 \in L_3$ is a partial translation of $s_2 \in L_2$ with respect to $\mathcal{A}_2$ and $s_2 \in L_2$ is a partial translation of $s_1 \in L_1$ with respect to $\mathcal{A}_1$.

Problem Statement

Let $\mathcal{A}_1$ be a semantic alignment of T_1 with T_2, underlying a semantic integration $\mathcal{I}_1$; let $\mathcal{A}_2$ be a semantic alignment of T_2 with T_3, underlying a semantic integration $\mathcal{I}_2$; let $\mathcal{A}$ be a semantic alignment of T_1 with T_3, underlying a semantic

integration $\mathcal{I}$.



What are sufficient and/or necessary conditions on $\mathcal{I}$, $\mathcal{I}_1$ and $\mathcal{I}_2$ for $\mathcal{A}$ to be a composition of $\mathcal{A}_1$ and $\mathcal{A}_2$?

4.2 Dynamic Ontology Refinement

Unlike with the previously discussed systems, the Ontology Refinement System (ORS) [McN06] does not match two ontologies by computing semantic relationships between ontology elements, but actually *refines* one of the ontologies —the ontology of a so called *planning agent*— in order to succeed in delegating a certain action to a foreign *service-providing agent* with slightly differing ontology. In addition it does not look for a refinement on the whole ontology, but only on the fragment whose mismatch caused the plan execution carried out by the planning agent to fail. Refinement is hence tightly linked with the agent's plan execution, and is solved by applying changes to the agent's ontology.

ORS is capable of diagnosing ontological mismatches that require the application of refinements. These mismatches occur, for instance, when, for a particular action α and in a particular situation s the service-providing agent is incapable of performing α as expected by the planning agent. This happens when, according to the planning agent's ontology O_{PA}, α's precondition should be satisfied in s but, according to the service-providing agent's ontology O_{SPA}. α's precondition is not satisfied in s. ORS refines O_{PA} in a way that α's precondition in O_{SPA} follows from O_{PA} in situation s. Refinements consist of replacing a predicate with another one that is above or below it in the predicate hierarchy of the ontology, adding or substracting predicate arguments, and including or removing rule preconditions, etc. These refinements change the theory or the signature of the ontology, or both.

Semantic Integration Revisited

In a sense ORS aims at a semantic integration of two ontologies, but not with respect to a reference theory, as in the previously discussed systems, but with respect to a situation and for a particular action. An ontology is a pair $O = \langle \mathsf{sign}(O), \mathsf{axs}(O) \rangle$, where $\mathsf{sign}(O)$ is a signature and $\mathsf{axs}(O)$ is the set of axioms of O over signature $\mathsf{sign}(O)$. Taking $Sen(\mathsf{sign}(O))$ to be the set of all first-order sentences over $\mathsf{sign}(O)$, an ontology O determines a theory $T_O = \langle Sen(\mathsf{sign}(O)), \vdash_{T_O} \rangle$, where $\vdash_{T_O}$ is the smallest first-order consequence relation such that $\emptyset \vdash_{T_O} \{\varphi\}$, for all $\varphi \in \mathsf{axs}(O)$. Consequently, semantic integration of ontologies reduces to semantic integration of theories. We may, hence, restate our definition of semantic integration given in Definition 1 as follows:

Definition 9 Two theories T_1 and T_2 are *semantically integrated with respect to a situation s and an action α*, if the following propositions are equivalent:

1. α's precondition in T_2 follows from T_1 in situation s

2. α's precondition in T_1 follows from T_2 in situation s

As with Definition 1, the definition above attempts to model the result of a semantic integration, such as ontology matching or ontology refinement. In ORS the latter is an asymmetric process, since only the planning agent's ontology O_{PA} is refined. Consequently, Definition 9 above can be relaxed (and this is how things actually work in ORS) by asking only for an implication 1. $\Rightarrow$ 2. (or else 2. $\Rightarrow$ 1.). Still, it is very ORS-specific: it is stated in terms of *situations* and *actions* as ORS performs semantic integration in the planning domain. It remains an open problem to find a more task-independent definition of semantic integration which still retains the core insight of semantic integration we get from Definitions 1 and 9, namely that semantic integration occurs always relative to some shared entity: a (virtual) reference theory on one hand, and particular situations and actions on the other hand.

Semantic Refinement vs. Semantic Alignment

Since ORS does not *align* ontologies as defined in Definition 5, it also remains an open problem to provide a faithful formalisation of ORS's refinements in terms of suitable mathematical objects. Still, we can provide a necessary condition that we expect ORS refinements to satisfy, namely that a refinement renders two theories to be semantically integrated for some situation and action:

Proposition 4 *Given two theories T_1 and T_2, if T_1' is a refinement of T_1 for T_2, then there exist a situation s and an action α such that T_1' and T_2 are semantically integrated with respect to s and α, while T_1 and T_2 are not.*

Obviously, semantic integration is understood here in terms of Definition 9.

4.3 An Interaction Model for Semantic Alignment

By exchanging *units of meaning coordination* two agents may progressively align their ontologies. For a practical application of the framework to ontology alignment in open, distributed environments, in this section we show how the framework of Section 2 serves as a foundation for a general ontology-alignment interaction model. We shall first describe the process of meaning coordination from an operational perspective, and then provide an executable specification of such interaction model by using LCC [Rob04].

The strategy that each agent may follow in selecting appropriate units of meaning coordinations will obviously influence the quality of the alignment that one eventually gets. In the process of meaning coordination we describe next, agents A_1 and A_2 alternate in exchanging units of meaning coordination (hereafter, UMC) in order to explain each other the meaning of local and foreign types. This process gradually builds up an alignment and is based on the following coordination tactic: if an agent A_i wants to known the meaning of a foreign concept, it asks agent A_j for an instance of this concept in order to classify this instance according to its own ontology; reciprocally, A_i may inform A_j which concept he has selected for this particular instance. This dialogue may be described schematically as follows:

Agent A_i wants to know the meaning of O_j-concept α:

1. A_j selects a new instance a for O_j-concept α
2. A_j sends A_i the UMC "a is an instance of concept α"
3. A_i selects an O_i-concept β for instance a
4. A_i sends A_j the UMC "a is an instance of concept β"

At this point, both A_i and A_j may update the alignment $\mathcal{A}$ as defined in Section 2.3 because the dialogue above involves the exchange of two UMCs. Also, A_j may take the new O_i-concept β as starting point for an analogous dialogue in order to find out the meaning of this foreign concept:

Agent A_j wants to know the meaning of O_i-concept β:

1. A_i selects a new instance b for O_i-concept β
2. A_i sends A_j the UMC "b is an instance of concept β"
3. A_j selects an O_j-concept γ for token b
4. A_j sends A_i the UMC "b is an instance of concept γ"

Again, at this point, both A_i and A_j may update the alignment $\mathcal{A}$ as defined in Section 2.3, because again the dialogue involves the exchange of two UMCs. Notice that this second dialogue is identical to the first one, only with the roles of agents A_i and A_j switched.

In the meaning coordination process described above we have been deliberately silent on how agents select instances and concepts for the UMCs they need to exchange, and also at the stage at which the alignment process finishes (e.g., because some good-enough alignment has been achieved). In an open, distributed system the strategy followed by agents will surely depend on the local decision-making machinery. Instead, we want to focus on the shared interaction model that agents would need to follow to coordinate their ontologies, independently of their particular decision-making strategies. For this, we need to supply, in an executable language, the specification of the general process of agent coordination that may yield (subject to the agents' strategies) to an alignment of ontologies, with the roles undertaken by the agents during that process. In the remainder of this section we give a specification of the meaning coordination described above in one such language, namely LCC [Rob04] (see Figure 1 for a definition of LCC's syntax).

Each of Clauses 1 to 3 defines the message-passing behaviour of a role in the interaction. Clause 1 defines the message-passing behaviour of an agent (identified by A_i) in the role of an *aligner* of ontologies. An agent in this role initiates a dialogue with another agent in the same role with the objective of building on top of an alignment $\mathcal{A}^n$ (which initially, for $n = 0$, may be empty) through the exchange of UMCs.

When in the role of an *aligner*, agent A_i either may choose to wait for a message from an agent A_j asking A_i to explain the meaning of a concept α, and switching subsequently to the role of an *explainer* of α for agent A_j; or else it may choose to send a message to an agent A_j asking A_j to explain the meaning

$$
\begin{array}{rcl}
\mathit{Interaction\_Model} & := & \{\mathit{Clause}, \ldots\} \\
\mathit{Clause} & := & \mathit{Agent} :: \mathit{Dn} \\
\mathit{Agent} & := & \mathbf{a}(\mathit{Role}, \mathit{Id}) \\
\mathit{Dn} & := & \mathit{Agent} \mid \mathit{Message} \mid \mathit{Dn}\ \mathbf{then}\ \mathit{Dn} \mid \mathit{Dn}\ \mathbf{or}\ \mathit{Dn} \mid \mathit{Dn}\ \mathbf{par}\ \mathit{Dn} \mid \mathbf{null} \leftarrow C \\
\mathit{Message} & := & M \Rightarrow \mathit{Agent} \mid M \Rightarrow \mathit{Agent} \leftarrow C \mid M \Leftarrow \mathit{Agent} \mid C \leftarrow M \Leftarrow \mathit{Agent} \\
C & := & \mathit{Term} \mid C \wedge C \mid C \vee C \\
\mathit{Role} & := & \mathit{Term} \\
M & := & \mathit{Term}
\end{array}
$$

Where **null** denotes an event which does not involve message passing; *Term* is a structured term (e.g., a Prolog term) and *Id* is either a variable or a unique identifier for an agent.

Figure 1: Syntax of LCC interaction models

of a concept α, and switching subsequently to the role of an *inquirer* of α for agent A_j.

$$
\begin{array}{l}
\mathbf{a}(\mathsf{aligner}(\mathcal{A}^n), A_i) :: \\
\quad \left(\begin{array}{l} \mathsf{ask}(\mathsf{explain}(\alpha)) \Leftarrow \mathbf{a}(\mathsf{aligner}(\_), A_j) \quad \mathbf{then} \\ \mathbf{a}(\mathsf{explainer}(\mathcal{A}^n, \alpha, A_j), A_i) \end{array} \right) \\
\quad \mathbf{or} \\
\quad \left(\begin{array}{l} \mathsf{ask}(\mathsf{explain}(\alpha)) \Rightarrow \mathbf{a}(\mathsf{aligner}(\_), A_j) \quad \mathbf{then} \\ \mathbf{a}(\mathsf{inquirer}(\mathcal{A}^n, \alpha, A_j), A_i) \end{array} \right)
\end{array}
\tag{1}
$$

Clause 2 defines the message-passing behaviour of an agent (identified by A_i) in the role of an *explainer* of a local concept α for agent A_j. An agent in this role exchanges with its dual agent (the agent that switched to the *inquirer* role when both initiated the dialogue in the *aligner* role) a pair of UMCs in order to update its alignment $\mathcal{A}^n$ with the ontology of A_j.

When in the role of an *explainer* of a local concept α for agent A_j, an agent A_i first sends a message to agent A_j (in the role of an *inquirer*) telling it that a is an instance of α, conditioned to A_i being capable of selecting such instance a for α. Next, it sends a new message to A_j asking it to classify a according to A_j's ontology. Then it waits for a message from A_j telling A_i that a is an instance of some foreign concept β. A_i then updates its current alignment $\mathcal{A}^n$ according to the exchanged UMCs (that a is and instance of concept α and of concept β), which yields the new alignment $\mathcal{A}^{n+1}$. Finally, A_i may choose to either continue the alignment with A_j, switching to the role of an inquirer of foreign concept β for agent A_j, or else it may choose to exit the dialogue by switching back to the initial *aligner* role with the updated alignment.

$$
\begin{array}{l}
\mathbf{a}(\mathsf{explainer}(\mathcal{A}^n, \alpha, A_j), A_i) :: \\
\quad \mathsf{tell}(\mathsf{is\_instance\_of}(a, \alpha)) \Rightarrow \mathbf{a}(\mathsf{inquirer}(\_, \_, \_), A_j) \leftarrow \mathsf{select\_instance}(\alpha, a) \quad \mathbf{then} \\
\quad \mathsf{ask}(\mathsf{classify}(a)) \Rightarrow \mathbf{a}(\mathsf{inquirer}(\_, \_, \_), A_j) \quad \mathbf{then} \\
\quad \mathsf{tell}(\mathsf{is\_instance\_of}(a, \beta)) \Leftarrow \mathbf{a}(\mathsf{inquirer}(\_, \_, \_), A_j) \quad \mathbf{then} \\
\quad \mathbf{null} \leftarrow \mathsf{update}(a, \alpha, \beta, \mathcal{A}^n, \mathcal{A}^{n+1}) \quad \mathbf{then} \\
\quad (\ \mathbf{a}(\mathsf{inquirer}(\mathcal{A}^{n+1}, \beta, A_j), A_i) \quad \mathbf{or} \quad \mathbf{a}(\mathsf{aligner}(\mathcal{A}^{n+1}), A_i)\ )
\end{array}
\tag{2}
$$

That is, given two agents in the aligner role of an interaction, when one asks the other for an explanation of a concept, the former will switch into the role of an inquirer (the one sending out the message asking for the explanation), while the latter will switch into the role of an explainer (the one getting the message). Explainer and inquirer agents then enter a dialogue in which they subsequently

exchange UMCs, until they decide (according to their local decision-making machinery) to exit the dialogue, an fall back to the initial aligner role. While in the explainer or inquirer role an agent will only pass messages with its dual agent.

Clause 3 defines the message-passing behaviour of an agent (identified by A_i) in the role of an of an *inquirer* of a foreign concept β for agent A_j. An agent in this role exchanges with its dual agent (the agent that switched to the *explainer* role when both initiated the dialogue in the *aligner* role) a pair of UMCs in order to update its alignment $\mathcal{A}^n$ with the ontology of A_j.

When in the role of an *inquirer* of a foreign concept β for agent A_j, an agent A_i first waits for a message of agent A_j (in the role of an *explainer*) telling it that b, for example, is an instance of β, and subsequently waits again for a new message from A_j that asks A_i to classify b according to A_i's ontology. It then sends a message to A_j telling it that b is an instance of local concept α, conditioned to A_i being capable of selecting such concept α for which b is an instance. Next, A_i updates its current alignment $\mathcal{A}^n$ according to the exchanged UMCs that b is an instance of concept α and of concept β), which yields the new alignment $\mathcal{A}^{n+1}$. Finally, A_i may choose to either continue the alignment with A_j, switching to the role of an explainer of local concept α for agent A_j, or else it may chose to exit the dialogue by switching back to the initial *aligner* role with the updated alignment.

$$
\begin{array}{l}
\mathbf{a}(\mathsf{inquirer}(\mathcal{A}^n, \beta, A_j), A_i) :: \\
\quad \mathsf{tell}(\mathsf{is\_instance\_of}(b, \beta) \Leftarrow \mathbf{a}(\mathsf{explainer}(\_, \_, \_), A_j) \quad \mathbf{then} \\
\quad \mathsf{ask}(\mathsf{classify}(b)) \Leftarrow \mathbf{a}(\mathsf{explainer}(\_, \_, \_), A_j) \quad \mathbf{then} \\
\quad \mathsf{tell}(\mathsf{is\_instance\_of}(b, \alpha) \Rightarrow \mathbf{a}(\mathsf{explainer}(\_, \_, \_), A_j) \leftarrow \mathsf{select\_concept}(b, \alpha) \quad \mathbf{then} \\
\quad \mathbf{null} \leftarrow \mathsf{update}(b, \alpha, \beta, \mathcal{A}^n, \mathcal{A}^{n+1}) \quad \mathbf{then} \\
\quad (\ \mathbf{a}(\mathsf{explainer}(\mathcal{A}^{n+1}, \alpha, A_j), A_i) \quad \mathbf{or} \quad \mathbf{a}(\mathsf{aligner}(\mathcal{A}^{n+1}), A_i)\ )
\end{array}
\tag{3}
$$

The three clauses above specify an executable interaction-model by which two agents align their ontologies by exchanging UMCs. Being independent of the classifying and decision-making machinery each agent might have, it offers a general model of ontology-alignment to which different agents can subscribe to.

5 Concluding Discussion

While research into ontology matching has produced increasingly complex algorithms, most settings in which the matching problem was tackled was almost always the same: given two ontologies, find all the possible mappings between their entities attaching a confidence level to the mappings that are returned. One of the challenges in the field of ontology matching now is not so much perfecting these algorithms, but rather trying to adapt them to novel scenarios. For instance, when integrating data from online ontologies it is often necessary to map between several online ontologies. This is very unlike the traditional scenario where only two ontologies were mapped at a time.

An example of mapping in such scenario is that performed in the context of the PowerAqua ontology-based question answering system [LMU06], where terms of the question will need to be dynamically mapped to several online ontologies. This run-time mapping brings up several challenges which need to be solved by the PowerMap mapping algorithm of PowerAqua [LSM06]. As a

result, PowerMap needs to be able to create mappings between heterogeneous data on-the-fly and with no predetermined assumption about the source and the ontological structure of these data. Rather than mapping being performed during the development of the application it now needs to be performed at run-time.

In terms of the framework presented in this deliverable PowerMap organises ontology mapping in three phases of increasing computational complexity: First, a syntactic mapping of terms to candidate ontologies is carried out. This step is analogous to the anchoring process described in Section 3.1, although PowerMap does not select a reference ontology a priori, as one might not know in advance which terms from which ontologies one may want to map. Next, semantic mapping is carried out on the previously computed syntactic mapping using the hierarchical structure of the ontologies and an external lexical source. Such semantic mapping is analogous to establishing a semantic integration with respect to a reference ontology such as WordNet, and uses techniques similar to the those described in Section 3.2. PowerMap, however, also exploits the semantics of the ontology's *is-a* relationships to obtain the meaning of a term. Finally, a semantic filtering step is done in order to filter out those ontologies that do not cover all terms of the question. Thus, in PowerMap the mapping process is driven by the task that has to be performed, more concretely by the query that is asked by the user. Semantic integration is hence relative to the particular question-answering problem. It is in this sense situation- and task-dependant in a similar fashion as described in Section 4.2.

We have discussed elsewhere [S+06] lengthily the need of extending the notion of ontology matching, as it has been understood in traditional applications, to dynamic ontology matching, and considered five general matching directions which we believe can appropriately address those requirements: approximate and partial ontology matching, interactive ontology matching, continuous "design-time" ontology matching, community-driven ontology matching and multi-ontology matching. In this deliverable we have explored this need from its mathematical foundations. We have presented a framework in which we gave concise definitions of semantic integration and alignment, and in which we were capable to describe current ontology matching technologies in a unifying manner. But this framework has also been suitable to highlight fundamental issues and open problems for semantic integration that arise in the context of P2P systems. We expect in OpenKnowledge to continue addressing these theoretical foundation of the semantic heterogeneity problem to devise theoretically sound semantic matching, alignment and refinement technology for open, distributed, P2P systems.

Acknowledgments. This work is supported under the OpenKnowledge[5] Specific Targeted Research Project (STREP), which is funded by the European Commission under contract number FP6-027253. The OpenKnowledge STREP comprises the Universities of Edinburgh, Southampton, and Trento, the Open University, the Free University of Amsterdam, and the Spanish National Research Council (CSIC).

[5] `http://www.openk.org`

References

[AKtv06] Zharko Aleksovski, Michel Klein, Warner ten Kate, and Frank van Harmelen. Matching unstructured vocabularies using a background ontology. In *15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks — EKAW 2006*, 2006.

[BS97] Jon Barwise and Jerry Seligman. *Information Flow: The Logic of Distributed Systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1997.

[CA03] Flávio Corrêa da Silva and Jaume Agustí. *Knowledge Coordination*. Wiley, 2003.

[CN00] Mihai Ciocoiu and Dana Nau. Ontology-based semantics. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR 2000, Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA, April 11-15, 2000*, pages 539–548. Morgan Kaufmann, 2000.

[DH01] J. Michael Dunn and Gary M. Hardegree. *Algebraic Methods in Philosophical Logic*. Oxford University Press, 2001.

[End02] Herbert Enderton. *A Mathematical Introduction to Logic*. Academic Press, second edition, 2002.

[FFR97] Adam Farquhar, Richard Fikes, and James Rice. The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46(6):707–727, 1997.

[GB92] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[Gog05] Jospeh Goguen. Information integration in institutions. To appear in a memorial volume for Jon Barwise edited by L. Moss. Draft available at http://www.cs.ucsd.edu/users/goguen/pps/ifi04.pdf, 2005.

[GSY05] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Semantic schema matching. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, pages 347–365. Springer, 2005.

[GW99] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis*. Springer, 1999.

[Ken00] Robert E. Kent. The information flow foundation for conceptual knowledge organization. In *6th International Conference of the International Society for Knowledge Organization*, Toronto, Canada, July 2000.

[Ken05] Robert E. Kent. Semantic integration in the information flow framework. In Yannis Kalfoglou, Marco Schorlemmer, Amit Sheth, Steffen Staab, and Michael Uschold, editors, *Semantic Interoperability and Integration*, volume 04391 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany, 2005.

[KS03] Yannis Kalfoglou and Marco Schorlemmer. IF-Map: An ontology-mapping method based on information-flow theory. In Stefano Spaccapietra, Sal March, and Karl Aberer, editors, *Journal on Data Semantics I*, volume 2800 of *Lecture Notes in Computer Science*, pages 98–127. Springer, 2003.

[Len95] Douglas Lenat. CyC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 1995.

[LMU06] Vanessa López, Enrico Motta, and Victoria Uren. PowerAqua: Fishing the semantic web. In *3rd European Semantic Web Conference — ESWC 2006*, 2006.

[LSM06] Vanessa López, Marta Sabou, and Enrico Motta. PowerMap: Mapping the semantic web on the fly. In *5th International Semantic Web Conference — ISWC 2006*, 2006.

[McN06] Fiona McNeill. *Dynamic Ontology Refinement*. PhD thesis, School of Informatics, The University of Edinburgh, 2006.

[Rob04] David Robertson. Multi-agent coordination as distributed logic programming. In Bart Demoen and Vladimir Lifschitz, editors, *Logic Programming*, volume 3132 of *Lecture Notes in Computer Science*, pages 416–430. Springer, 2004.

[S+06] Pavel Shvaiko et al. Dynamic ontology matching: a survey. Deliverable D3.1, OpenKnowledge STREP FP6-027253, May 2006.

[SK05] Marco Schorlemmer and Yannis Kalfoglou. Progressive ontology alignment for meaning coordination: An information-theoretic foundation. In Frank Dignum et al., editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, July 25–29, 2005*, pages 737–744. ACM Press, 2005.

[vBD+05] J. van Diggelen, R. J. Beun, F. Dignum, R.M. van Eijk, and J.-J. Ch. Meyer. Combining normal communication with ontology alignment. In *AAMAS 2005 International Workshop on Agent Communication*, 2005.

[WG02] Jun Wang and Les Gasser. Mutual online ontology alignment. In *OAS'02 Ontologies in Agent Systems, Proceedings of the AAMAS 2002 Workshop*, volume 66 of *CEUR Workshop Proceedings*, 2002.

[ZKEH06] Antoine Zimmermann, Markus Krötzsch, Jérôme Euzenat, and Pascal Hitzler. Formalizing ontology alignment and its operations with category theory. In *International Conference on Formal Ontology in Information Systems — FOIS 2006*, 2006.