Conjunctive Queries for Ontology based Agent Communication in MAS

Cassia Trojahn
Universidade de Evora
Rua Romao Ramalho, 59
Evora, Portugal
cassia@di.uevora.pt

Paulo Quaresma
Universidade de Evora
Rua Romao Ramalho, 59
Evora, Portugal
pq@di.uevora.pt

Renata Vieira
Pontificia Universidade
Catolica do Rio Grande do Sul
Av. Ipiranga, 6681
Porto Alegre, Brazil
renata.vieira@pucrs.br

ABSTRACT

In order to obtain semantic interoperability in open Multi-Agent Systems, agents need to agree on the basis of different ontologies. In this paper we formally define mapping as correspondences between queries over ontologies. Individual mappings are computed by specialized agents using different mapping approaches. Next, these agents use argumentation to exchange their local results, in order to agree on the mappings. Based on their preferences and *strength* of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments. These arguments are then represented as conjunctive queries in OWL-DL extended with DL-safe rules [9], a restriction imposed to attain decidability in such query answering system.

1. INTRODUCTION

Multi-Agent Systems are by nature distributed and heterogeneous. In these systems, ontologies play a fundamental role, formalizing the vocabulary from the agent's perception of the world. In open Multi-Agent Systems, such as the Web, agents using different ontologies need to agree on the vocabulary they use, in order to communicate and then resolve their tasks. Ontology mapping is a primary problem that has to be solved in order to allow agents with different backgrounds to adjust themselves before starting any form of cooperation or communication. Using a common ontology is impractical, because it would result in assuming a standard communication vocabulary and it does not take into account the conceptual requirements of agents that could appear in future [13]. Moreover, a common ontology forces an agent to abandon its own world view and adopt one that is not specifically designed for its task [4].

In this paper we propose an OWL-DL mapping system based on argumentation and conjunctive queries. First, individual mappings are computed by specialized agents using different mapping approaches (lexical, semantic and structural). Next, these agents use argumentation to exchange their local results, in order to agree on the obtained mappings. An Extended Value-based Argumentation Framework (E-VAF) is used to represent arguments with *strength* [24]. The E-VAF allows to determine which arguments are acceptable, with respect to the different *audiences* represented by different agents. To each argument is associated a *strength*, representing how confident an agent is in the similarity of two ontology terms. Based on their preferences and confidence of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments.

We follow the framework of [15] where the mappings obtained from argumentation are expressed as correspondences between queries over ontologies. Query answering in such a mapping system of this general form is undecidable and requires a theorem prover. To attain decidability in such query answering, we use the restrictions from [9]. These restricted, but still very expressive mappings, can be expressed in OWL-DL extended with DL-safe rules. Query answering in such framework is based on a technique for reducing OWL-DL ontologies to disjunctive datalog programs [12][9]. Our motivation for using DL conjunctive queries is to improve the mapping expressiveness and consequently the application efficiency.

This paper is structured as follows. Section 2 presents our Extended Value Argumentation Framework (E-VAF). Section 3 presents the use of E-VAF for ontology mapping. Section 4 presents an evaluation of it against three mapping systems. Section 5 presents the proposed mapping system based on conjunctive queries for OWL-DL ontologies. Section 6 presents an illustrative example. Section 7 comments on related work. Finally, section 8 presents the final remarks and the future work.

2. EXTENDED VALUE-BASED ARGUMENTATION FRAMEWORK

Our argumentation model is based on the Value-based Argumentation Framework (VAF)[3], a development of the classical argument system of Dung [6]. The VAF is able to distinguish attacks from successful attacks, those which defeat the attacked argument, with respect to an ordering on the values that are associated with the arguments. It allows to accommodate different audiences with different interests and preferences. We extend the VAF in order to represent arguments with *strength*, which represents the *confidence degree* that an agent has in some argument. One element has been added to VAF: a function which maps from arguments to *strength*. We assumed that *strength* is a relevant criterion to represent the ontology mapping domain. In previous work

Cite as: Conjunctive Queries for Ontology based Agent Communication in MAS, Trojahn, Cássia and Quaresma, Paulo and Vieira, Renata, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.),May,12-16.,2008,Estoril,Portugal,pp. 829-836.

[24] we had used only two discrete classes to express the *confidence degree* an agent had in the mappings (*certainty* and *uncertainty*). In this paper, we use *confidence* as *strength* ∈ [0,1]. This work differs from previous ones in three issues: confidence degrees are represented as a continuous values (previously they were discrete); terms with multi-words are treated, an important issue for ontology mapping not considered in previous work; representing mappings obtained from argumentation as conjunctive queries is original, in the sense of a complete framework for ontology mapping.

Definition 2.1 An Extended Value-based Argumentation Framework (E-VAF) is a 7-tuple *E-VAF = (AR, attacks,V,val,valS)* where *(AR,attacks,V,val, P)* is a value-based argumentation framework, and *valS* is a function which maps from elements of AR to real values from [0,1].

Definition 2.2 An argument $x \in AR$ defeats$_a$ (or *successfully attacks*) an argument $y \in AR$ for audience a if and only if *attacks(x,y)* $\wedge$ *((valS(x) > valS(y))* $\vee$ ($\neg$ *valpref(val(y),val(x))* $\wedge$ ($\neg$ *(valS(y) > valS(x)))*).

An attack succeeds if (a) the *strength* of the attacking argument is greater than the *strength* of the argument being attacked; or if (b) the argument being attacked does not have greater preference value than attacking argument (or if both arguments relate to the same preference values) and the *strength* of the argument being attacked is not greater than the attacking argument.

Definition 2.3 A set S of arguments is *conflict-free* for audience a if *(∀x)(∀y) ((x ∈ S ∧ y ∈ S)* $\longrightarrow$ *(¬attacks(x, y)* $\vee$ *(¬(valS(x) > valS(y))* $\wedge$ *(valpref(val(y), val(x))* *(∨ (valS(y) > valS(x))))))*.

3. E-VAF FOR ONTOLOGY MAPPING

In this paper we consider three *values*: lexical (L), semantic (S), and structural (E) (i.e. $V = \{L, S, E\}$, where $V \in$ S-VAF). These values represent the mapping approach used by the agent and are also used to represent the audiences. Each audience has an ordering preference between the *values*. For instance, the lexical agent represents an audience where the value L is preferred to the values S and E. Our idea is not to have an individual audience with preference between the agents (i.e., semantic agent is preferred to the other agents), but it is to try accommodate different audiences (agents) and their preferences.

We point out that agent autonomy is not the main focus of the work, the contribution relies more on the use of argumentation for solving the difficult problem of maximizing correct conceptual mapping on the basis of conflicting individual views.

3.1 Argumentation generation

First, the agents work in an independent manner, applying the mapping approaches and generating mapping sets. The mapping result will consist of a set of correspondences between terms of two Description Logic ontologies (OWL-DL, as commented in Section 5). For efficiency and private or proprietary reasons, the mappings can be computed only over the terms the agents need to understand each other, rather than over the whole ontology.

A mapping m can be described as a 3-tuple $m = (t_1,t_2,h)$, where t_1 corresponds to a term in the ontology 1, t_2 corresponds to a term in the ontology 2, and h is one of {+,-} depending on whether the argument is that m does or does not hold. An argument is defined as follows:

Definition 3.1 An *argument* $\in AR$ is a 3-tuple $x = (m,a,s)$, where m is a mapping; $a \in V$ is the value of the argument (lexical, semantic or structural); s is the *strength* of the argument.

3.1.1 Lexical agent

The lexical agent adopts a metric to compare string similarity. We used the *lexical similarity* proposed by [19]. This metric is based on the Levenshtein distance [16], which is given by the minimum number of operations (insertion, deletion, or substitution of a single character) needed to transform one string into another. The length of the compared terms is considered to compute the *lexical similarity*. This metric returns a value from the interval [0,1], where 1 indicates high similarity between two terms.

Differently from the previous work [24][25], the agents are able to deal with compound terms. The first step in this process is the tokenization, where the terms are parsed into tokens by a tokenizer. The *strength* of an argument is computed according to the *lexical similarity* between each token of the two compared terms. Table 1 shows the possible values to s and h, where t_{Sn} correspond to some token of the source term (source ontology), and t_{Tn} correspond to some token of the target term (target ontology). Two tokens are *lexically similar* if the *lexical similarity* is greater than a *threshold r*.

When all tokens are *lexically similar* with each other, the terms match and the *strength* of the argument is 1. In this case, for instance, the lexical agent generates an argument x = (m,L,1), where m = (t_1,t_2,+).

If *some* tokens of the terms are *lexically similar*, the *strength* is computed according to the number of tokens that matches, according to the *calc-s* formula, where T_S is the term from the source ontology, T_T is the term from the target ontology, and nM is the number of tokens that match between T_S and T_T:

$$calc\text{-}s = max\left(0, \frac{max(\mid T_S \mid, \mid T_T \mid) \; - \; nM)}{max(\mid T_S \mid, \mid T_S \mid)}\right)$$

If there are no lexically similar tokens between the terms, the agent is not sure that the terms map (i.e., *strength* equals to 0), because this agent knows that other agent can resolve this mapping. In the specific case, if there is no lexical similarity between the terms, the semantic agent can resolve that mapping.

3.1.2 Semantic agent

This agent considers the semantic relations (i.e., synonym, hyponym, and hypernym) between concepts to measure the similarity between them. WordNet[1] database, a large repository of English semantically related items, is used to provide these relations. Table 2 shows the possible values to s and h according to the semantic similarity.

When all tokens have semantic relation with each other, the *strength* of the argument is 1. The agent generates,

[1]http://www.wordnet.princeton.edu

Table 1: h and s to lexical audience.

s	+ (h)
1	t_{S1} lexically similar to t_{T1} t_{S1} lexically similar to all t_{T1}, ..., t_{Tn} all t_{S1}, ..., t_{Sn} lexically similar to t_T all t_{S1}, ..., t_{Sn} lexically similar to all t_{T1}, ..., t_{Tn}
calc-s	t_{S1} lexically similar to some t_{T1}, ..., t_{Tn} some t_{S1}, ..., t_{Sn} lexically similar to t_T some t_{S1}, ..., t_{Sn} lexically similar to some t_{T1}, ..., t_{Tn}
s	- (h)
0	t_{S1} no lexically similar a_{T1} t_{S1} no lexically similar to all t_{T1}, ..., a_{Tn} all t_{Sn}, ..., t_{Sn} no lexically similar to t_{S1} all t_{S1}, ..., t_{Sn} no lexically similar to all t_{T1}, ..., t_{Tn}

Table 2: h and s to semantic audience.

s	+ (h)
1	t_{S1} relation with t_{T1} t_{S1} relation with all t_{T1}, ..., t_{Tn} all t_{S1}, ..., t_{Sn} with relation with t_T all t_{S1}, ..., t_{Sn} with relation with all t_{T1}, ..., t_{Tn}
calc-s	t_{S1} some relation with some t_{T1}, ..., t_{Tn} some t_{S1}, ..., t_{Sn} relation with t_T some t_{S1}, ..., t_{Sn} relation with some t_{T1}, ..., t_{Tn}
s	- (h)
0	t_{S1} no relation with t_{T1} t_{S1} no relation with all t_{T1}, ..., t_{Tn} t_{S1}, ..., t_{Sn} no relation with t_T t_{S1}, ..., t_{Sn} no relation with t_{T1}, ..., t_{Tn}

for instance, an argument x = (m,S,1), where m= (t_1,t_2, +). If *some* tokens have semantic relation, the *strength* is computed according to the number of semantically related tokens (formula presented above). Otherwise, if there are no semantic relation between the tokens, the agent is not sure that the terms map (i.e., *strength* equals to 0), because this agent knows that other agent can resolve the mapping. In the specific case, when the searched terms are not available in WordNet, the lexical agent can decide the mapping. It is common because there is no complete lexical database for every domain (i.e., WordNet is incomplete for some domains).

3.1.3 Structural agent

The structural agent considers the positions of the terms in the ontology hierarchy to verify if the terms can be mapped. First, it is verified if the super-classes of the compared terms are lexically similar. If not, the semantic similarity is used. For instance, if the super-classes of the terms are not lexically similar, but they are synonymous, an argument x = (m,E,s), where m = (t_1,t_2,+), is generated, where s varies according to the rules from Tables 1 or 2.

However, there are two main differences among the *strength* returned by the lexical, semantic, and structural agents. As Table 1 and Table 2, when the agents can not resolve the mapping, the *strength* of the corresponding argument is 0. However, if the structural agent does not find similarity (lexical or semantic) between the super-classes of the compared terms, it is because the terms can not be mapped (i.e., the terms occurs in different contexts). Then, the *strength* for no mapping is 1. Otherwise, if the structural agent finds similarity between the super-classes of the compared terms, it is because they can be mapped, but it does not mean that the terms have lexical or semantic similarity, then the *strength* for the mapping is 0. For instance, for the terms "Publication/Topic" and "Publication/Proceedings", the structural agent indicates that the terms can be mapped because they have the same super-class, but not with *strength* 1 because it is no able to indicate that the terms are similar. Otherwise, for the terms "Digital-Camera/Accessories" and "Computer/ Accessories", the agent can indicate that the terms can not be mapped because they occur in different contexts (*no-mapping* with *strength* equal to 1).

3.2 Preferred extension generation

After generating their set of arguments, the agents exchange with each other their arguments. When all agents have received the set of arguments of each other, they generate their *attacks* set. An *attack* (or counter-argument) will arise when we have arguments for the mapping between the same terms, but with conflicting values of h. For instance, an argument x = (m_1,L,+) has as an *attack* an argument y = (m_2,E,-), where m_1 and m_2 refer to the same terms in the ontologies. The argument y also represents an *attack* to the argument x.

As an example, consider the mapping between the terms "Subject" and "Topic" and lexical and semantic values. The lexical agent generates an argument x = (m,L,0), where m = (subject$_S$,topic$_T$,-); and the semantic agent generates an argument y = (m,S,1), where m = (subject$_S$,topic$_T$,+). For both lexical and semantic audiences, the set of arguments is AR= {x,y} and the *attacks* = {(x,y),(y,x)}.

When the set of arguments and attacks have been produced, the agents need to define which of them must be accepted. To do this, the agents compute their preferred extension, according to the audience's preferences and *strength*. A set of arguments is *globally subjectively acceptable* if each element appears in the preferred extension for some agent. A set of arguments is *globally objectively acceptable* if each element appears in the preferred extension for every agent. The arguments which are neither objectively nor subjectively acceptable are considered *indefensible*.

In the example above, considering the lexical(L) and semantic(S) audiences, where L $\succ$ S and S $\succ$ L, respectively, for the lexical audience, the argument y successfully attacks the argument x, while the argument x does not successfully attack the argument y for the semantic audience. Then, the preferred extension of both lexical and semantic agents is composed by the argument y, which can be seen as globally *objectively* acceptable.

4. ARGUMENTATION EVALUATION

Let us consider that three agents need to obtain a consensus about mappings that link corresponding class names in two different ontologies. We used the *Test 8*[2] data, which corresponds to ontologies of company profiles domain. The source and target ontologies in *Test 8* have 10 and 16 classes, respectively, resulting 160 possible mappings. The terms are composed from 1 to 5 tokens (for instance "Oil-and-Gas-Exploration-and-Production" or "Petroleum-Product- Distribution").

We used these ontologies because the comparative results for them, using three mapping systems commented below, are available in [8]. The three mapping systems are: Cupid[17], COMA[5], and S-Match[8]. Cupid algorithm is based on linguistic and structural approaches. It matches individual schema elements based on their names, data types, domains, etc. The result is a linguistic similarity, *lsim*, between each pair of elements. Next, structural matching of schema elements is made, which depends in part on linguistic matches calculated initially, resulting in a structural similarity coefficient, *ssim*. The weighted similarity (*wsim*) is a mean of *lsim* and *ssim*: wsim = wstruct ssim + (1−wstruct) lsim, where the constant wstruct is in the range 0 to 1.

COMA represents a generic system to combine match results. The match result is a set of mapping elements specifying the matching schema elements together with a similarity ∈ [0,1] indicating the plausibility of their correspondence. The matchers currently supported fall into three classes: simple, hybrid and reuse-oriented matchers. They exploit different kinds of schema information, such as names, data types, and structural properties, or auxiliary information, such as synonym tables and previous match results.

S-Match algorithm is based on two main steps. First, the synonymous terms are captured using WordNet (element level). Second, the structural schema properties are taken into account, where the path to the root is computed (structural level). Element level semantic matchers provide the input to the structural level matcher, which is applied on to produce the set of semantic relations between concepts as the matching result.

Table 3 shows the comparative results. We used a threshold of 0.8 to lexical agent classifies the mappings (terms with lexical similarity greater than 0.8 are considered similar) and a threshold to eliminate the terms that have *strength* below 0.75. Our model returned better precision than Cupid and COMA, and equal precision when compared to S-Match (precision equal to 1). When comparing the f–measure values, our model had similar result than COMA and S-Match and better result than Cupid.

Differently from these works, our model uses argumentation to combine map- ping approaches. Although our implementation does not provide the best solution for the ontology mapping problem for these experimental tests as yet, we claim that our main contribution is to propose a model that can be used to combine different approaches. Using argumentation has the following advantages: the agents are independent to each other; many other agents can be easily added to our model, without having to modify the implementation; there are several techniques for ontology mappings, which can be adapted according to domain, kind of ontologies, and available resources (for instance, in the context of some languages, there is no lexical databases such WordNet). Moreover, although the reported experiments consider pairs of ontologies, the approach can be easily extended to deal with a larger number of ontologies. Also, the use of web-based background knowledge could be easily integrated in the framework as another mapping agent.

[2]http://dit.unitn.it/~accord/Experimentaldesign.html

5. REPRESENTING MAPPINGS AS OWL-DL CONJUNCTIVE QUERIES

The mappings obtained from argumentation are represented as conjunctive queries in OWL-DL extended with DL-safe rules [9], a restriction imposed to attain decidability in such query answering system. In the following, we first introduce OWL-Description Logic (OWL-DL) and then comment on the representation of mappings as conjunctive queries.

5.1 OWL-DL

Description Logics (DLs) is the name for a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain ("the world") by defining the relevant concepts of the domain (its terminology), and using these concepts to specify properties of objects and individuals occurring in the domain [1].

A DL knowledge base (KB) comprises two components: TBox and ABox. The TBox contains the *terminology*, which specifies the vocabulary of an application domain. The ABox contains assertions about named individuals in terms of the TBox. The vocabulary consists of *concepts* and *roles*. *Concepts* denote set of individuals while *roles* denote binary relationship between individuals. Atomic concepts and roles can be used to build complex description of concepts and roles, using *constructors*. The language for building descriptions is a feature of different DLs, and different systems are distinguished by their description languages, i.e., the expressiveness of the language according with the *constructors* that they support.

The OWL-DL ontology language is a variant of *SHOIN(D)* [10] Description Logic, which provides constructors for full negation, disjunction, a restricted form of existential quantification, and reasoning with concrete datatypes. OWL-DL is the state-of-the-art formalism to represent ontologies. Here, we propose to use OWL-DL also to represent the mappings obtained from argumentation. The set of *SHOIN(D)* concepts is defined by the following syntactic rules, where A is an atomic concept, R is a role name, d is a concrete domain, c_i are individuals, and n is a non-negative integer:

$$
\begin{array}{lcl}
\mathrm{C} & \rightarrow & \mathrm{A} \mid \neg\mathrm{C} \mid \mathrm{C}_1 \sqcap \mathrm{C}_2 \mid \mathrm{C}_1 \sqcup \mathrm{C}_2 \mid \exists\mathrm{R}.\mathrm{C} \mid \forall\mathrm{R}.\mathrm{C} \mid \\
 & & \mathrm{n\ S} \mid \mathrm{n\ S} \mid \{\mathrm{a}_1,...,\mathrm{a}_n\} \mid \mathrm{n\ T} \mid \mathrm{nT} \mid \\
 & & \exists \mathrm{T}_1, ..., \mathrm{T}_n.\mathrm{D} \mid \forall \mathrm{T}_1, ..., \mathrm{T}_n.\mathrm{D} \\
\mathrm{D} & \rightarrow & \mathrm{d} \mid \{\mathrm{c}_1,...,\mathrm{c}_n\ \}
\end{array}
$$

Here, we consider the semantics of a *SHOIN(D)* knowledge base KB by the mapping π proposed by [9] which transforms KB axioms into a first-order formula. Each atomic concept is mapped into a unary predicate and each role is mapped into a binary predicate.

5.2 Conjunctive Queries

Definition 5.1 (Conjunctive Queries) [9]. Let KB be a *SHOIN(D)* knowledge base, and N_P be a set of predicate symbols, such that all *SHOIN(D)* concepts and

Table 3: Comparative mapping results.

Ontology	Arg			Cupid			COMA			S-Match		
	P	R	F	P	R	F	P	R	F	P	R	F
Company profiles (160)	1	0.63	0.77	0.50	0.60	0.54	0.80	0.70	0.74	1.0	0.65	0.78

all roles are in N_P. An atom has the form P(s_1, ..., s_n), often denoted as P(s), where P $\in N_P$, and s_i are either variables or individuals from KB. An atom is called ground atom, if it is variable-free. An atom is called a DL-atom if P is a *SHOIN(D)* concept or role.

Consider that x_1, ..., x_n and y_1, ..., y_n are sets of distinguished and non distinguished variables, indicated as x and y. A conjunctive query over KB – Q(x,y) – is a conjunction of atom $\bigwedge P_i(s_i)$, where all s_i together contain x and y.

A conjunctive query Q(x,y) is DL-safe if each variable occurring in a DL-atom also occurs in a non-DL-atom in Q(x,y): $\pi(Q(x,y)) = \exists y: \bigwedge \pi(P_i(s_i))$. For $Q_1(x,y_1)$ and $Q_2(x,y_2)$ conjunctive queries, a query containment axiom $Q_1(x,y_1) \sqsubseteq Q_2(x,y_2)$ has the following semantics: $\pi(Q_2(x,y_2)) \sqsubseteq Q_1(x,y_1) = \forall x: \pi((Q_1(x,y_1)) \leftarrow Q_2(x,y_2)$.

The main inferences for conjunctive queries are:

- Query answering. An answer of a conjunctive query Q(x,y) w.r.t. KB is an assignment θ of individuals to distinguished variables, such that $\pi(KB) \models \pi(Q(x\theta,y))$.
- Checking query containment. A query $Q_2(x,y_2)$ is contained in a query $Q_1(x,y_1)$ w.r.t. KB, if $pi(KB) \models \pi(Q_2(x,y_2)) \sqsubseteq Q_1(x,y_1)$.

Intuitively, DL-safely restricts the applicability of a query only to individuals explicitly named in a KB [9]. To automatically convert a non-DL safe query into a DL-safe one, it is assumed a special non-DL predicate *O* such that, for each individual α occurring in KB, it contains a fact *O(x)*. Then, a non-DL safe conjunctive query Q(x,y) can be converted into a DL-safe query by appending to it an atom of the form *O(z)*, for each variable *z* occurring only in a DL-atom of Q(x,y).

5.3 OWL-DL Mapping System

Following the definitions from [15] and [9], we introduce an OWL-DL mapping system based on conjunctive queries. The components of mapping system are the source ontology, the target ontology, and the mapping between the two ontologies.

Definition 5.2 (OWL-DL Mapping System) An OWL-DL mapping system *MS* is a triple (S,T,M), where

- S is the source ontology;
- T is the target ontology;
- M is the mapping between S and T, i.e, a set of assertions $q_S \rightsquigarrow q_T$, where q_S and q_T are conjunctive queries over S and T, respectively, with the same set of distinguished variables x, and $\rightsquigarrow$ *in* $\{\sqsubseteq, \sqsupseteq, \equiv\}$.

An assertion $q_S \sqsubseteq q_T$ is called a *sound mapping*, requiring that q_S is contained by q_T w.r.t S $\cup$ T; an assertion $q_S \sqsupseteq q_T$ is called a *complete mapping*, requiring that q_T is contained by q_S w.r.t S $\cup$ T; and an assertion $q_S \equiv q_T$ is called an *exact mapping*, requiring it to be sound and complete. A sound mapping $q_S \sqsubseteq q_T$ is equivalent to an axiom $\forall x: q_T(x,y_T) \leftarrow q_S(x,y_S)$, while a complete mapping $q_T \sqsubseteq q_S$ is equivalent to an axiom $\forall x: q_S(x,y_S) \leftarrow q_T(x,y_T)$.

Definition 5.3 (Mapping System Semantics) For a mapping system MS = (S,T,M), let $\pi(MS) = \pi(S) \cup \pi(T)$ $\pi(M)$. the main inference for MS is computing answers of Q(x,y) w.r.t. MS, for Q(x,y) a conjunctive query.

The intuitive reading of this semantics is that an answer of a query needs to be entailed by the source ontology S, the target T and the mappings M. Query answering in such a system of this general form is undecidable and requires a theorem prover. Here we use the special kinds of mappings introduced by [9], that lead to decidable query answering and for which practical query answering algorithms exist:

- *Full Implication Mappings*: The first class of mappings captures the mappings that can be directly expressed in OWL-DL. This is the case if q_s and q_t are DL-atoms of the form $P_s(x)$ and $P_t(x)$.
  - Concept Mappings. If q_s and q_t are of the form $P_s(x)$ and $P_t(x)$ and P_s and P_t are DL concepts, the mapping corresponds to the equivalent concept inclusion axiom.
  - Role mappings. If q_s and q_t are of the form $P_s(x_1,y_1)$ and $P_t(x_1,y_2)$, with P_s and P_t roles, the mapping corresponds to the equivalent role inclusion axiom.
- *Restricted Implication Mappings*: Query answering for general implication mappings is undecidable due to the unrestricted use of non distinguished variables in either q_s or q_t. In the following, restrictions that reduce the expressivity of the mappings but provide for a decidable query answering procedure are defined.
  - *DL-safe Mappings.* Consider a mapping $q_S \sqsubseteq q_T$ with the assertions $\forall x: q_T(x,y_T) \leftarrow q_S(x,y_S)$. In order to avoid introducing new objects in the interpretation domain, it is not allowed the use of non-distinguished variables in the query q_T, i.e., restrict the assertions to the form $\forall x: q_T(x) \leftarrow q_S(x,y_S)$. Query answering with such mappings is still undecidable in the general case. Therefore, it is required the query q_S to be DL-safe, thus limiting the applicability of the rules to known individuals.
  - *Mappings with Tree-like Query Parts.* The restrictions introduced by DL-safety may appear rather strong. In the following it is commented how to relax the above restrictions for a certain class of so-called tree-likes queries. Using the query roll-up technique from [11], we can eliminate non-distinguished variables by reducing a tree-like part of a query to a concept, without

loosing semantic consequences. By applying rolling up to each role term, an arbitrary query can be reduced to an equivalent one which contains only concept terms, and which can be answered using a set of satisfiability tests in DL.

5.4 Mappings as Conjunctive Queries

The mappings M are obtained using the E-VAF, as commented in Section 3. The set of globally (or subjectively) acceptable arguments is taken as input to the process of creating the conjunctive queries. A conjunctive query, as commented in Section 4, has the form $\bigwedge (P_i(s_i))$. Here, each $P_i(s_i)$ corresponds to a mapping. For instance, in the example showed in Section 3.2, Q(x): Subject(x) $\equiv$ Topic(x).

Using conjunctive queries in DL allows to represent very expressive mappings. When a term T_S is mapped with more than one T_T, the query has, for instance, the form Q(x): Thesis(x) $\equiv$ (PhdThesis $\sqcup$ MasterThesis)(x). According to the restrictions for decidable queries imposed by [9], x must be defined in the ABox (DL-safe rules).

Considering roles and concepts, we can define complex mapping such as Q_1(x,y): Publication(x) $\sqcup$ title(x,y) and Q_2(x,y): Entry(x) $\sqcup$ entry-title(x,y).

Table 4 shows how to compute answers to a conjunctive query Q(x,y). Here, we adapt the algorithm proposed by [9]. Query answering in a DL KB is reduced to query answering in a Disjunctive Datalog KB, which can be performed efficiently, using the techniques of (disjunctive) deductive databases [12]. The algorithm starts by eliminating non-distinguished variables from Q(x,y) and the mapping, using the query roll-up technique [11] (i.e., transforming roles into concept descriptions). After roll-up, the obtained mappings and queries are required to be DL-safe, which is needed for decidable query answering. If this condition is fulfilled, then the source ontology, target ontology and the mappings are converted into a disjunctive datalog program, and the query is answered in the obtained program.

Table 4: Algorithm for Answering Queries

Require: Mappings M in a conjunctive query format, ontology source S and ontology targetT
1: Roll-up tree-like parts of Q(x,y)
2: Roll-up tree-like parts of query mapping in M
3: Stop if Q(x,y) or some mapping from M is not DL-safe
4: KB $\leftarrow S \cup T \cup M$
6: Reduce KB to a Disjunctive Datalog KB (DD)
5: Compute the answer of Q(x,y) in DD

6. AN ILLUSTRATIVE EXAMPLE

Let us consider an example where two agents, *Initiator*(I) and *Participant*(P), interact with each other to agree on the price of a product, using FIPA Contract Net Interaction Protocol. We had used two ontologies about product schemas[3].

Figure 1 shows the organizational model of the system, and Table 5 describes the steps of the interaction between the agents.

The agent *Initiator* has the ontology presented in Table 6 and the *Participant* has the ontology presented in Table 7. Comparing the output of our argumentation model

[3]http://dit.unitn.it/~accord/Experimentaldesign.html (Test 4)

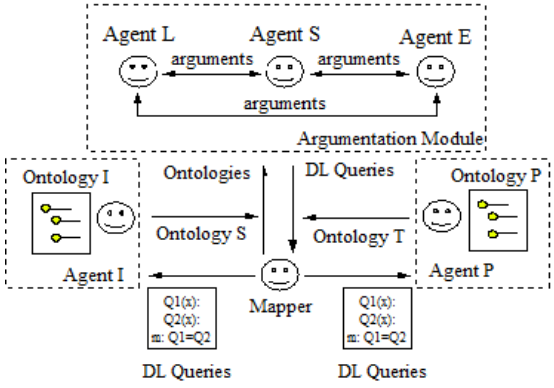

Figure 1: Organizational model.

and manual mapping, it obtained precision equal to 1, recall equal to 0.75, and f–measure equal to 0.85. Table 8 shows the arguments and counter-arguments (attacks, *At.*), for the correct mappings returned by our model.

Table 6: Initiator's DL Ontology.

Consumer-Electronics $\sqsubseteq \top$
Personal-Computers $\sqsubseteq$ Consumer-Electronics
Microprocessors $\sqsubseteq$ Personal-Computers
Accessories $\sqsubseteq$ Personal-Computers
Photo-and-Cameras $\sqsubseteq$ Consumer-Electronics

As shown in Table 8, the preferred extensions of the agents are composed by the arguments generated by the corresponding audience. The preferred extension of the lexical agents is {1, 5, 7}; the preferred extension of the semantic agent is {2, 5, 8}; and the preferred extension of the structural agent is {3, 6, 9}). Here, we consider the *"subjectively acceptable"* arguments. So, the arguments 1, 2, 3, 5, 6, 7, and 8 can be considered as consensus (note that the argument 4 is not acceptable). Note that the arguments 1, 2 and 3; 5 and 6; and 7, 8 and 9 refer to the same mappings. The corresponding queries are shown in Table 9.

Let us consider an extension of the first query in order to represent complex mappings with attributes: Q_{1I}(x,y): Photo-and-Camera(x) $\sqcup$ name(x,y) $\sqcup$ price(x,z) and Q_{1P}(x,y): Camera-and-Photo(x) $\sqcup$ name-camera(x,y) $\sqcup$ price-camera (x,z), with $Q_{1I} \equiv Q_{1P}$.

Figure 2 shows an AUML interaction diagram with the messages exchanged between the agents P and I during the negotiation of the price of one camera. The agents use the queries to search for correspondences between the messages sent from each other and the terms in the corresponding ontologies. In the example, the agent I sends a message to the agent P, using its vocabulary. The agent P, then, converts the message, using the DL queries.

Table 7: Participant's DL Ontology.

Electronics $\sqsubseteq \top$
PC $\sqsubseteq$ Electronics
PC-board $\sqsubseteq$ PC
Cameras-and-Photo $\sqsubseteq$ Electronics
Accessories $\sqsubseteq$ Cameras-and-Photo
Digital-Cameras $\sqsubseteq$ Cameras-and-Photo

Table 5: Interaction steps.

Step	Description
1	Mapper (M) agent requests the ontologies to be mapped to agents I and P
2	Ontologies are sent from M to the argumentation module
3	Agents using lexical (L), semantic (S) and structural (E) approaches apply their algorithms
4	Agents L, S and E communicate with each other to exchange their arguments
5	Preferred extensions of L, S and E are generated
6	Subjectively arguments are computed
7	Mappings are represented as conjunctive queries
8	Queries are sent to Mapper
9	Queries are sent to I and P
10	Agents I and P use the queries to communicate with each another

Table 8: Arguments and attacks.

ID	Argument	At.
1	(Photo-and-Camera,Camera-and-Photo,+,L,1.0)	-
2	(Photo-and-Camera,Camera-and-Photo,+,S,1.0)	-
3	(Photo-and-Camera,Camera-and-Photo,+,E,0.0)	-
4	(Personal-Computer,PC,-,L,0.0)	5, 6
5	(Personal-Computer,PC,+,S,1.0)	4
6	(Personal-Computer,PC,+,E,0.0)	4
7	(Consumer-Electronic,Electronic,+,L,0.5)	-
8	(Consumer-Electronic,Electronic,+,S,0.5)	-
9	(Consumer-Electronic,Electronic,+,E,0.0)	-

Table 9: Conjunctive queries.

Query ID	Correspondences
$Q_{I1}(x)$	i:Photo-and-Camera(x)
$Q_{P1}(x)$	p:Camera-and-Photo(x)
m_1	$Q_{I1} \equiv Q_{P1}$
$Q_{I2}(x)$	i:Personal-Computer(x)
$Q_{P2}(x)$	p:PC(x)
m_2	$Q_{I2} \equiv Q_{P2}$
$Q_{I3}(x)$	i:Consumer-Electronic(x)
$Q_{P3}(x)$	p:Electronic(x)
m_3	$Q_{I3} \equiv Q_{P3}$

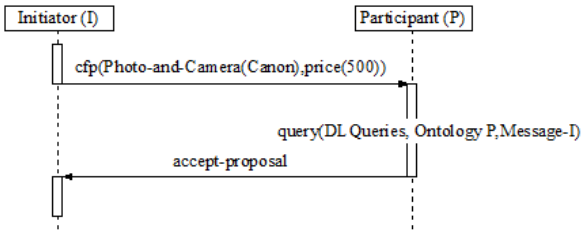

Figure 2: AUML negotiation interaction.

7. RELATED WORK

In the field of ontology mapping, [20], [21], and [7] present a broad overview of the various approaches on automated ontology matching. Expressivity of the mappings is an issue that is not well explored in these works. [9] was the first to propose expressive mapping as DL queries. However, the mappings are not obtained by the framework proposed, only represented as queries. Our proposal is an integrated framework which combines the representation of mappings as queries, as proposed by [9], and a complete DL mapping system based on argumentation.

In the field of ontology argumentation few approaches are being proposed. Basically, the closer proposal is from [14][13], where an argument framework is used to deal with arguments that support or oppose candidate correspondences between ontologies. The candidate mappings are obtained from an Ontology Mapping Repository (OMR) – the focus is not how the mappings are computed – and argumentation is used to accommodate different agent's preferences. In our approach mappings are computed by the specialized agents described in this paper, and argumentation is used to solve conflicts between the individual results.

We find similar proposals in the field of ontology negotiation. [23] presents an ontology to serve as the basis for agent negotiation, the ontology itself is not the object being negotiated. A similar approach is proposed by [4], where agents agree on a common ontology in a decentralized way. Rather than being the goal of each agent, the ontology mapping is a common goal for every agent in the system. [2] presents an ontology negotiation model which aims to arrive at a common ontology which the agents can use in their particular interaction. We, on the other hand, are concerned with delivering mapping pairs found by a group of agents using argumentation. [22] describes an approach for ontology mapping negotiation, where the mapping is composed by a set of semantic bridges and their inter-relations, as proposed in [18]. The agents are able to achieve a consensus about the mapping through the evaluation of a confidence value that is obtained by utility functions. According to the confidence value the mapping rule is accepted, rejected or negotiated. Differently from [22], we do not use utility functions. Our model is based on cooperation and argumentation, where the agents change their arguments and by argumentation they select the preferred mapping.

8. FINAL REMARKS AND FUTURE WORK

In this paper we proposed to represent ontology mappings as correspondences between queries. The mappings are computed by agents using different mapping approaches. These agents exchange their local results and use argumentation to agree on the obtained mappings. An Extended Value-based Argumentation Framework (E-VAF) was used to represent arguments with *strength*. Based on their preferences and *strength* of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments. These arguments are then represented as conjunctive queries in OWL-DL.

Although we have not emphasized the use of mapping for agent communication, we took it for granted that ontology mapping is intrinsically related to agent communication issues. This is a primary problem that has to be solved in

order to allow agents with different backgrounds to adjust themselves before starting any form of cooperation or communication.

We evaluated our argumentation model against three mapping systems, according to the available comparative data and we presented through an example how to represent the mapping results using conjunctive queries. The motivation for using such queries is for improved efficiency. However, the improvement was not yet demonstrated, just the decidability. It is an ongoing research work.

In the future, we intend to develop further tests considering a benchmark of ontologies[4]; demonstrate the efficiency of using conjunctive queries to search in an integrated ontology system; and use the mapping as input to an ontology merge process in the question answering domain.

Acknowledgments

The first author is supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E05D059374BR.

9. REFERENCES

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[2] S. Bailin and W. Truszkowski. Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*, 17(1):7–19, 2002.

[3] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.

[4] J. v. Diggelen, R. Beun, F. Dignum, v. R. Eijk, and J. C. Meyer. Anemone: An effective minimal ontology negotiation environment. In *Proceedings of the Fiftheen International Conference on Autonomous Agents and Multi-Agent Systems*, pages 899–906, 2006.

[5] H. H. Do and E. Rahm. Coma - a system for flexible combination of schema matching approaches. In *Proceedings of the 28th Conference on Very Large Databases*, 2002.

[6] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n–person games. *Artificial Intelligence*, 77:321–358, 1995.

[7] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[8] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: An algorithm and an implementation of semantic matching. In *Proceedings of the European Semantic Web Symposium*, pages 61–75, 2004.

[9] P. Haase and B. Motik. A mapping system for the integration of owl-dl ontologies. In *IHIS*, pages 9–16, 2005.

[10] I. Horrocks and P. Patel-Schneider. Reducing owl entailment to description logic satisfiability. In *Proceedings of the 2nd International Semantic Web Conference*, 2003.

[11] I. Horrocks and S. Tessaris. A conjunctive query language for description logic aboxes. In *AAAI/IAAI*, pages 399–404, 2000.

[12] U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ description logic to disjunctive datalog programs. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning*, pages 152–162. AAAI Press, 2004.

[13] L. Laera, I. Blacoe, V. Tamma, T. Payne, J. Euzenat, and T. Bench-Capon. Argumentation over ontology correspondences in mas. In M. Durfee, E. H.; Yokoo, editor, *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2007.

[14] L. Laera, V. Tamma, J. Euzenat, T. Bench-Capon, and T. R. Payne. Reaching agreement over ontology alignments. In *Proceedings of 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[15] M. Lenzerini. Data integration: a theoretical perspective. In *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246, New York, NY, USA, 2002. ACM Press.

[16] I. Levenshtein. Binary codes capable of correcting deletions, insertions an reversals. In *Cybernetics and Control Theory*, 1966.

[17] J. Madhavan, P. Bernstein, , and E. Rahm. Generic schema matching with cupid. In *Proceedings of the Very Large Data Bases Conference*, pages 49–58, 2001.

[18] A. Maedche, B. Motik, N. Silva, and R. Volz. Mafra - a mapping framework for distributed ontologies. In *13th International Conference on Knowledge Engineering and Knowledge Management*, pages 235–250, 2002.

[19] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management*, pages 251–263, 2002.

[20] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB*, 10:334–350, 2001.

[21] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. Technical report, Informatica e Telecomunicazioni, University of Trento, 2004.

[22] N. Silva, P. Maio, and J. Rocha. An approach to ontology mapping negotiation. In *Proceedings of the K-CAP Workshop on Integrating Ontologies*, 2005.

[23] V. Tamma, M. Wooldridge, I. Blacoe, and I. Dickinson. An ontology based approach to automated negotiation. In *Proceedings of the IV Workshop on Agent Mediated Electronic Commerce*, pages 219–237, 2002.

[24] C. Trojahn, P. Quaresma, and R. Vieira. A cooperative approach for composite ontology mapping. *LNCS Journal of Data Semantic (to appear)*, 2007.

[25] C. Trojahn, P. Quaresma, and R. Vieira. An extended value-based argumentation framework for ontology mapping with confidence degrees. In *Workshop of Argumentation, International Conference on Autonomous Agents and Multiagent Systems*, 2007.

[4] http://oaei.ontologymatching.org/