# A Cooperative Approach for Composite Ontology Mapping

Cássia Trojahn[1], Márcia Moraes[2], Paulo Quaresma[1] and Renata Vieira[3]

[1]Departamento de Informática, Universidade de Évora, Portugal
[2]Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Brazil
[3]Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos, Brazil
cassia@di.uevora.pt, mmoraes@pucrs.br, pq@di.uevora.pt, renatav@unisinos.br

**Abstract.** This paper proposes a cooperative approach for composite ontology mapping. We first present an extended classification of automated ontology matching and propose an automatic composite solution for the matching problem based on cooperation. In our proposal, agents apply individual mapping algorithms and cooperate in order to change their individual results. We assume that the approaches are complementary to each other and their combination produces better results than the individual ones. Next, we compare our model with three state of the art matching systems. The results are promising specially for what concerns precision and recall. Finally, we propose an argumentation formalism as an extension of our initial model. We compare our argumentation model with the matching systems, showing improvements on the results.

## 1 Introduction

Ontology mapping is the process of linking corresponding terms from different ontologies. The mapping result can be used for ontology merging, agent communication, query answering, or for navigation on the Semantic Web.

Different approaches to the matching problem have been proposed in the literature, see for example [28][30] and [31] for a taxonomy of the past approaches. We consider that to achieve high matching accuracy for a large variety of schemas, a single technique is unlikely to be successful [7].

We consider that different agents working on the basis of particular approaches arrive to distinct matching results that must be shared, compared, chosen and agreed. In order to deal with this problem, we present a composite mapping approach based on cooperative agents, which negotiate on a final matching result. We compare our model with three state of the art schema-based matching systems, namely Cupid[19], COMA[7], and S-Match[14]. The results are promising specially for what concerns precision and recall.

To deal with some mapping conflicts, which are not resolved by our negotiation model, we propose an argument formalism for composite ontology mapping.

We extend a state of art argumentation framework, namely the Value-based Argumentation Framework (VAF)[3], in order to represent arguments with confidence degrees. The VAF allows to determine which arguments are acceptable, with respect to the different *audiences* represented by different agents. We then associate to each argument a confidence degree, representing the confidence that a specific agent has in that argument.

In our novel proposal, cooperative agents apply individual mapping algorithms and cooperate in order to change theirs local results (arguments). Next, based on their preferences and confidence of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred mapping sets are viewed as the set of globally acceptable arguments. This is a more formal presentation for composite mapping. We also compare our argumentation model with the Cupid, COMA, and S-Match systems. The results are better than when using our negotiation model.

The paper is structured as follows. The next section briefly reviews the state of the art in ontology mapping. Section 3 comments on cooperative negotiation. Section 4 presents our negotiation model. Section 5 presents the results using the negotiation model. Section 6 presents the argumentation formalism and section 7 presents our argumentation model. Section 8 compares the results of the argumentation model and previous approaches. In section 9, related work are commented. Finally, section 10 presents the final remarks and the future work.

## 2 Ontology Mapping Approaches

The previous work of [28], [30] and [31] present a broad overview of the various approaches on automated ontology matching, classifying the mapping approaches in terms of input and techniques utilized in the mapping process. We propose a revision of the classification of mapping approaches presented in previous work, and we complement their proposals, including new elements in these classification. We point out that [20] presents other style of ontology mapping classification that is based on frameworks, methods and tools, translators, mediators, etc. We are not including these aspects in the classification presented here.

[28] distinguishes between individual and combining matchers. Individual matchers comprise schema-based and instance-based, element and structure levels, linguistic and constrained-based matching techniques. Combining matchers comprise hybrid and composite matchers.

Based on this previous taxonomy, [30] distinguishes between heuristic and formal techniques at schema-level; and implicit and explicit techniques at element- and structure-level. [31] introduces new criterias which are based on (i) general properties of matching techniques, i.e., approximate and exact techniques; (ii) interpretation of input information, i.e., syntactic, external, and semantic techniques at element and structure levels; and (iii) the kind of input information, i.e., terminological, structural, and semantic techniques.

Moreover, [13] distinguishes between weak semantics and strong semantics element-level techniques. Weak semantics techniques are syntax-driven techniques (e.g., techniques which consider labels as strings, or analyze data types, or soundex of schema elements) while strong semantics techniques exploit, at the element level, the semantics of labels (e.g., based on the use of thesaurus).

We present a revised classification in Figure 1 (our modifications are in bold font). As in [28], we distinguish between individual and combining matchers. However, we divided the individual matchers on data level, ontology level, or context level, but we kept the combining matcher divided on hybrid or composite.

At the data level, data instances are used as input to the matching process. At the ontology level, the terms of the ontology structure and the hierarchy are taking into account. Then, as [28], we distinguish between element-level matcher and structure level matcher. Finally, the ontology's application context can be used, i.e, how the ontology entities are used in some external context. This is specially interesting, for instance, to identify WordNet sense that must be considered to specific terms.

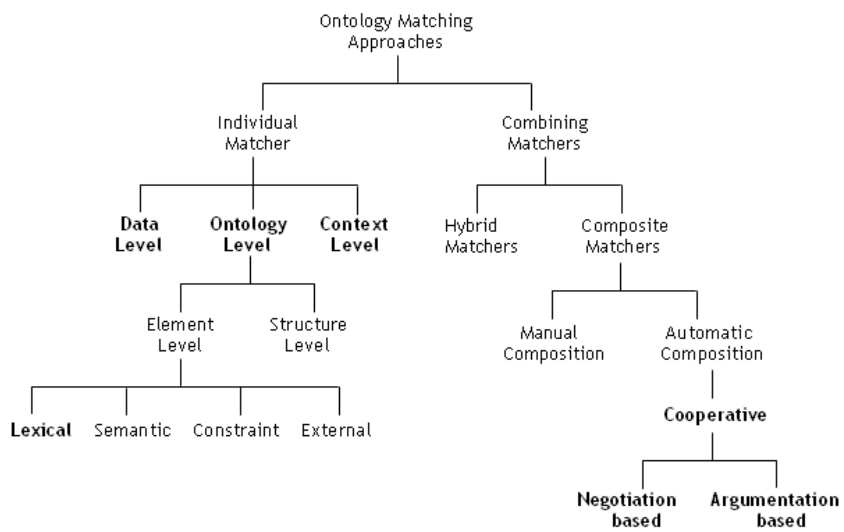


**Fig. 1.** Our classification of matching approaches.

At the element-level we consider, according to [31], semantic and external matchers. However, we replaced the syntactic by lexical and added a constraint-based matchers. We assume that the term "syntactic" refers to morpho-syntactic categories of words (i.e., implicating some word annotation). We consider that the term "lexical" is more appropriated to refer to the category of approaches based on string similarity.

The lexical approaches use metrics to compare string similarity. One well-known measure is the Levenshtein distance or edit distance [23], which is given by the minimum number of operations (insertion, deletion, or substitution of a single character) needed to transform one string into another. Based on Levenshtein measure, [25] proposes a lexical similarity measure for strings, the String Matching (SM), that considers the number of changes that must be made to change one string into the other and weighs the number of these changes against the length of the shortest string of these two. Other common metrics are: the Smith-Waterman[34], which additionally uses an alphabet mapping to costs; and the [11] which searches for the largest common substring.

Semantic matchers consider semantic relations between concepts to measure the similarity between them, usually on the basis of one thesaurus or similar semantic oriented linguistic resources. The well-known WordNet[1] database, a large repository of English items, has been used to provide these relations. This kind of mapping is complementary to the pure string similarity metrics. Cases where string metrics fail to identify high similarity between strings that represent completely different concepts are common. For example, for the words "score" and "store" the Levenshtein metric returns 0.68, which is a high metric if we consider that the they represent very different concepts. On the other hand terms like "student" and "learner" are semantically similar although they are lexically distant from each other.

Constraint-based matchers are based on data types, value ranges, uniqueness, cardinalities, and other information constraints in the matching process. For example, the similarity between two terms can be based on the equivalence of data types and domains, of key characteristics (e.g., unique, primary, foreign), or relationship cardinality (e.g., 1:1 relationships) [28].

Finally, at the element-level, we consider that external matchers consider some type of external information, such as user input or previous matching results.

Structural matchers use the ontology structure as input to the matching process (i.e., the positions of the terms in the ontology hierarchy are considered). Several approaches using this intuition have been proposed: super(sub)-concept rules consider that if super or sub concepts are the same, the actual concepts are similar to each other ( [5][10]); bounded path matching takes two paths with links between classes defined by the hierarchical relations, compare terms and their positions along these paths, and identify similar terms (see, for instance, Anchor-prompt algorithm [27][16]); leaves-rules, where two non-leaf schema elements are structurally similar if their leaf sets are highly similar, even if their immediate children are not, see, for example[19].

We also consider, as [28], hybrid and composite matchers, at combining matcher level. Hybrid matchers use multiple matching criteria (e.g., name and type equality) within an integrated matcher; and composite matchers (which can use a manual or automatic process) combine multiple match results produced by different match algorithms. Our approach is an automatic composite matcher

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

and then we add a cooperative approach at automatic level, which can be based on negotiation or argumentation. We point out that an automatic mapping approach can be also based on machine learning techniques, as presented by [8], which combines multiple matchers using a learning approach.

Due to the complexity of the problem using only one approach is usually not satisfactory. These approaches are complementary to each other. Combining different approaches must reflect a better solution when compared to the solutions of individual approaches. Our first proposal is to use a cooperative negotiation model, where agents apply individual mapping algorithms and negotiate on a final mapping result.

## 3 Cooperative Negotiation

Negotiation is a process by which two or more parties make a joint decision [38]. It is a key form of interaction that enables groups of agents to arrive at mutual agreement regarding beliefs, goals or plans [2]. Hence the basic idea behind negotiation is reaching a consensus [15].

Negotiation usually proceeds in a series of rounds, with every agent making a proposal at each round [37]. The process can be described as follow, based on [22]. One agent generates a proposal and other agents review it. If some other agent does not like the proposal, it rejects the proposal and might generate a counter-proposal. If so, the other agents (including the agent that generated the first proposal) review the counter-proposal and the process is repeated. It is assumed that a proposal becomes a solution when it is accepted by all agents.

Cooperative negotiation is a particular kind of negotiation where agents cooperate and collaborate to obtain a common objective. In cooperative negotiation, each agent has a partial view of the problem and the results are put together via negotiation trying to solve the conflicts posed by having only partial views [12].

This kind of negotiation has been currently adopted in resource and task allocation fields [4][26][38]. In these approaches, the agents try to reach the maximum global utility that takes into account the worth of all their activities. In our approach the cooperative negotiation is a form of interaction that enables the agents to arrive to mutual agreement regarding the result of different ontology mapping approaches.

## 4 Cooperative Negotiation Model for Composite Ontology Mapping

In our model, the agents use lexical, semantic and structural approaches to map terms of two different ontologies. The distinct mapping results are shared, compared, chosen and agreed, and a final mapping result is obtained. This approach aims to overcome the drawbacks of the using individual ontology mapping approaches. First, we present the organization of the agent society and next we detail the negotiation process.

### 4.1 Organization of the agent society

We describe our model according to an agent society (Figure 2), using the Moise+ model [18]. This model proposes three dimensions for the organization of agent societies: structural, functional and deontic. The structural dimension defines what agents could do in their environment (theirs roles). The functional dimension defines how agents execute their goals. The deontic dimension defines the permissions and obligations of a role in a goal. This paper focuses on the first dimension.

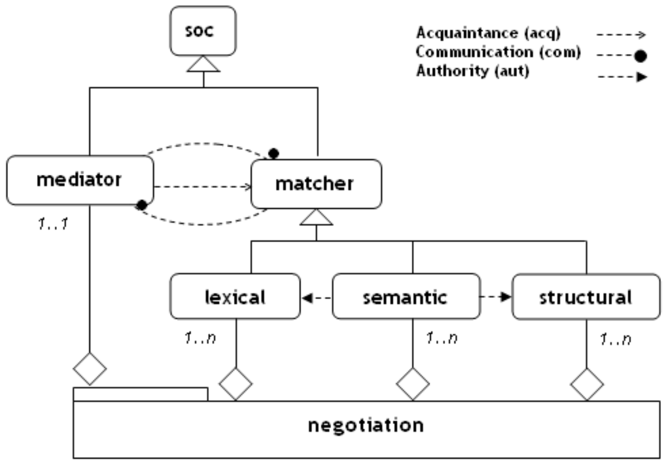


**Fig. 2.** Organizational model.

According to [18] and [17], structural specification has three main concepts, roles, role relations and groups that are used to build, respectively, the individual, social and collective structural levels of an organization. The individual level is composed by the roles of the organization. A role means a set of constraints that an agent ought to follow when it accepts to play that role in a group. The following roles are identified in the proposed organization:

- Mediator: this role is responsible for mediating the negotiation process, sending and receiving messages to and from the mapping agents.
- Matcher: this role is responsible for giving an output between two ontology mappings (i.e., encapsulates the mapping algorithms). One matcher could assume the lexical, semantic or structural role. On the lexical role, the matcher makes the mapping using algorithms based on string similarity. On the semantic role, the agent search by corresponding terms in a semantic oriented linguistic database. On the structural role, the agent is based on the intuition that if super-classes are the same, the compared classes are similar to each other. If sub-classes are the same, the compared classes are also similar.

In the social level are defined the kinds of relations among roles that directly constrain the agents. Some of the possible relations are:

– Acquaintance (acq): agents playing a source role are allowed to have a representation of the agents playing the destination role. In Figure 2, this kind of relation is present between the source role mediator and the destination role matcher.
– Communication (com): agents playing a source role are allowed to communicate with agents that play the destination role. In Figure 2 this kind of relation is present between the source role mediator and the destination role matcher (by heritage, lexical, semantic and structural).
– Authority (aut): agents playing a source role has authority upon agent playing destination role. In Figure 2 this kind of relation is present between the source role semantic and the destination roles lexical and structural.

The collective level specifies the group formation inside the organization. A group is composed by the roles that the system could assume, the sub-groups that could be created inside a group, the links (relations) valid for agent and by the cardinality. A group can have intra-groups links and inter-groups links. The intra-group links state that an agent playing the link source role in a group is linked to all agents playing the destination role in the same group or in its sub-groups. The inter-group links state that an agent playing the source role is linked to all agents playing the destination role despite the groups these agents belong to [18]. Links intra-group are represented by a hatched line and links inter-groups are represented by a continue line. This specification defines only a group called negotiation and all links are intra-group.

Based on the structural specification of the proposed organization, our society is composed by one agent that assumes the mediator role and three agents that assume the matcher role. One of the matcher agents is assuming the lexical role, one is assuming the semantic role, and one is assuming the structural role.

### 4.2 Negotiation process

Basically, the negotiation process involves two phases. First, the agents work in an independent manner, applying a specific mapping approach and generating a set of negotiation objects. A negotiation object is a 3-tuple $O = (t_1, t_2, C)$, where $t_1$ corresponds to a term in the ontology 1, $t_2$ corresponds to a term in the ontology 2, and $C$ is the mapping category resulting from the mapping for these two terms. Second, the set of negotiation objects, that compose the mapping is negotiated among the agents. The negotiation process involves one mediator and several matcher agents.

In order to facilitate the negotiation process (i.e, reduce the number of negotiation rules), we define four mapping categories according to the output of the matcher agents. Table 1 shows the categories and the corresponding mapping results.

**Lexical agent** The output of the lexical agents is a value from the interval [0,1], where 1 indicates high similarity between two terms (i.e, the strings are identical). The Levenshtein metric is used. For example, the words "reference" and "citation" have a Levenshtein value equals to 0.0. This way, if the output is 1, a "mapping with certainty" is obtained. If the output is 0, the agent has a "not mapping with certainty". A threshold is used to classify the output in uncertain categories. The threshold value is specified by the user.

**Semantic agent** The semantic agents consider semantic relations between terms according to the WordNet database. Relations such as synonym, antonym, holonym, meronym, hyponym, and hypernym can be returned for a given pair of terms. For instance, the semantic agent searches the relations between the terms "reference" and "citation" in the WordNet database and can assume that these terms are synonymous. Synonymous terms are considered as mapping with certainty; terms related by holonym, meronym, hyponym, or hypernym are considered mapping with uncertainty; when the terms can not be related by the WordNet (the terms are unknown for the WordNet database), the terms are considered as not mappings with uncertainty.

**Structural agent** The structural agent uses the super-classes intuition to verify if the terms can be considered similar. First, it is verified if the super-classes are lexically similar. Otherwise, the semantic similarity is used. If the super-classes are lexically or semantically similar, the terms are similar to each other. For instance, when mapping the terms "reference/thesis" (where "reference" is the super-class of "thesis") and "citation/proceeding", the structural agent indicates that the terms can be mapped because the super-classes are semantically similar. The matching category corresponds the output of the lexical or semantic comparison (e.g, if super-classes are not lexically similar, but they are considered synonymous, a "mapping with certainty" is returned).

We point out that semantic and structural mappings are complex problems, and in this paper we simple adopted state-of-art semantic and structural approaches. Therefore, we are composing on what is now generally available. We consider that using richer semantic and structural mappings is relevant, but our emphasis for this paper is in combining state of art approaches.

**Table 1.** Mapping categories.

| Category | Lexical | Semantic |
|---|---|---|
| Mapping (certainty) | 1 | synonym |
| Mapping (uncertainty) | $1 > r > t$ | related |
| Not mapping (uncertainty) | $0 < r <= t$ | unknown |
| Not mapping (certainty) | 0 | |

Figure 3 shows an AUML interaction diagram with the messages changed between the agents during a negotiation round. We use an extension of AUML-2 standard to represent agents' actions (the actions are placed centered over the lifeline of the named agent). The interaction diagram refers to negotiation of the mapping between the classes "personal computer " and "pc" (Figures 4 and 5)[2].

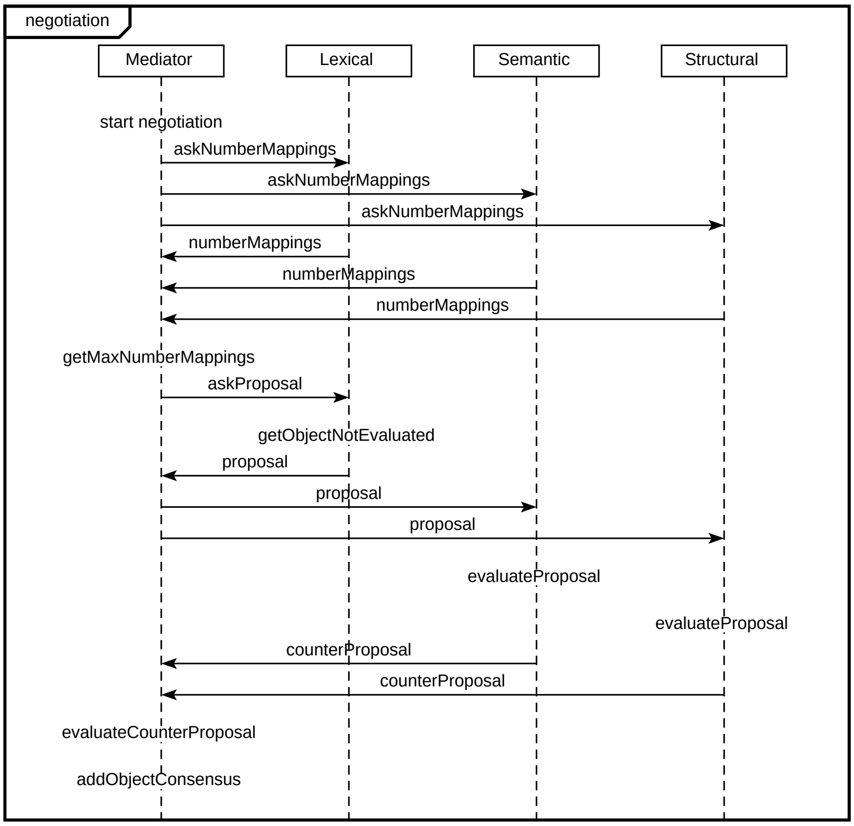


**Fig. 3.** AUML negotiation interaction.

The negotiation process starts with the mediator agent asking to the matcher agents for its number of "mappings with certainty". The first matcher agent to generate a proposal is one that has the greatest number of "mappings with certainty" (lexical agent, in the specific example).

The proposal contains the first negotiation object that still wasn't evaluated by the agent. This proposal is then sent to the mediator agent, which sends it to other agents (in the specific example, the lexical agent proposes a "not mapping with certainty" to the mapping between the classes "personal computer

[2] Ontologies available in http://dit.unitn.it/ accord/Experimentaldesign.html(Test 4)

```
<owl:Class rdf:ID="Electronics"> </owl:Class>

<owl:Class rdf:ID="Personal_Computers">
  <rdfs:subClassOf rdf:resource="#Electronics"/>
</owl:Class>

<owl:Class rdf:ID="Microprocessors">
  <rdfs:subClassOf rdf:resource="#Personal_Computers'
</owl:Class>

<owl:Class rdf:ID="Photo_and_Cameras">
  <rdfs:subClassOf rdf:resource="#Eletronics"/>
</owl:Class>

<owl:Class rdf:ID="Accessories">
  <rdfs:subClassOf rdf:resource="#Microprocessors"/>
</owl:Class>
```

**Fig. 4.** Ontology 1.

```
<owl:Class rdf:ID="Electronic"> </owl:Class>

<owl:Class rdf:ID="PC">
  <rdfs:subClassOf rdf:resource="#Electronic"/>
</owl:Class>

<owl:Class rdf:ID="PC_board">
  <rdfs:subClassOf rdf:resource="#PC"/>
</owl:Class>

<owl:Class rdf:ID="Camera_and_Photo">
  <rdfs:subClassOf rdf:resource="#Electronic"/>
</owl:Class>

<owl:Class rdf:ID="Accessory">
  <rdfs:subClassOf rdf:resource="#Camera_and_Photo"/>
</owl:Class>

<owl:Class rdf:ID="Digital_Camera">
  <rdfs:subClassOf rdf:resource="#Camera_and_Photo"/>
</owl:Class>
```

**Fig. 5.** Ontology 2.

" and "pc"). Each agent then evaluates the proposal, searching for an equivalent negotiation object. One negotiation object is equivalent to another when both refers to same terms which are being compared in the two ontologies.

If an equivalent negotiation object has the same category, the agent accepts the proposal. Otherwise, if the agent has a different category for the compared terms in the negotiation object, its object negotiation is sent as a counter-proposal to the mediator agent, which evaluates the several counter-proposals received (several agents can send a counter-proposal). In the example, semantic and structural agents have generated counter-proposals, indicating a "mapping with certainty" between the compared terms. The semantic agent identifies that the terms are synonymous in WordNet, and structural agent identifies terms having the same super-class (electronics).

The mediator selects one counter-proposal that has the greater number of hits. If two categories receive equals number of hits, the category indicated by the semantic agent is considered as the negotiation consensus. When a proposal is accepted by all agents or a counter-proposal consensus is obtained, the mediator adds the corresponding negotiation object in a consensus negotiation set and the matcher agents mark its equivalent one as evaluated. The negotiation ends when all negotiation objects are evaluated.

## 5 Experiments using the Negotiation Model

We applied our negotiation model to link corresponding class names in two different ontologies. The results produced by our negotiation model were compared with manual matches[3] (expert mappings). The manual matches specified

[3] Obtained from http://dit.unitn.it/ accord/Experimentaldesign.html

between the attributes of the ontologies were not considered in this set of experiments.

Previous experiments using our negotiation model were presented in [36]. This current work extends that previous one in many aspects. First, there we used only lexical and semantic agents in the negotiation process. Second, the resulting mapping category was obtained by majority, where the semantic agent had authority over the lexical agents (when two mapping categories received the equal number of hits, the semantic agent decides the resulting mapping category). Third, we used only two other ontologies related to bibliography domain to evaluate that initial proposal.

The negotiation model was implemented in Java for Windows, version 1.5.0, and the experiments ran on Pentium(R) 4, UCP 3.20GHz, 512MB. The lexical agent was implemented using the edit distance measure (Levenshtein measure). We used the algorithm available in the API for ontology alignment (INRIA)[4] (EditDistNameAlignment). The semantic agent uses the JWordNet API[5], which is an interface to the WordNet database. For each WordNet synset, we retrieved the synonymous terms and considered the hypernym, hyponym, member-holonym, member-meronym, part-holonym, and part-meronym as related terms. The structural agent is based on super-classes similarity.

The threshold used to classify the matcher agents output was 0.6. A pre-processing step was made, where special (e.g., _) and stop words (e.g., "and", "or", "of") were removed.

We have used four groups of ontologies: parts of Google and Yahoo web directories[6], product schemas[7], course university catalogs[8], and company profiles[9]. We considered the "mappings with certainty" and the "mappings with uncertainty" as examples of the positive classes. As a mapping quality measure, the well-know measures of precision, recall, and f–measure were used.

First, we compared the results obtained from our model with the results from expert mapping (Table 2 – the column "Others" contains mappings identified as corrects by our model, which where not identified by the experts). We also indicated the number of terms for each group of ontologies (only class names).

The negotiation consensus identified correctly all mappings defined by the expert, for all groups – all mappings defined by the expert were returned as "mappings with certainty" by our model. When considering the other mappings ("Others"), for the "Google and Yahoo", 3 "mappings with certainty" and 5 "mappings with uncertainty" have been returned. For instance, a "mapping with uncertainty" between the terms "/Arts/Visual_Arts" (where "Arts" is the super-class of "Visual_Arts") and "/Arts_Humanities/Design_ Art" has seen identified. This mapping was not defined by expert, however it could be considered as cor-

[4] http://alignapi.gforce.inria.fr
[5] http://jwn.sourceforge.net (using WordNet 2.1)
[6] http://dit.unitn.it/ãccord/Experimentaldesign.html (Test 3)
[7] http://dit.unitn.it/ãccord/Experimentaldesign.html (Test 4)
[8] http://dit.unitn.it/ãccord/Experimentaldesign.html (Test 7)
[9] http://dit.unitn.it/ãccord/Experimentaldesign.html (Test 8)

rect. This kind of "mapping with uncertainty" has been observed in the other examples. In "Product schemas", only one new mapping has been returned, being a "mapping with certainty", but incorrectly (i.e., "/Electronics/Personal _Computers/Accessories" and "/Electronic/Cameras_and_Photos/Accessories"). Finally, for the "Course catalogs", 3 new mappings were categorized as "mappings with uncertainty" (e.g., "/Courses/College_of_ engineering" and "/Courses/ College_of_Arts_and _Sciences").

**Table 2.** Expert mapping and consensus results.

| | | Consensus | |
|---|---|---|---|
| Ontology | Expert mapping | Correct | Others |
| Google and Yahoo directories (54) | 4 | 4 | 8 |
| Product schemas (30) | 4 | 4 | 1 |
| Course catalogs (48) | 6 | 6 | 3 |
| Company profiles (9) | 3 | 3 | 0 |

Second, we compared the output of all agents (Table 3). Using lexical or structural individual agents was not sufficient to obtain all corrects mappings. These agents did not classify correctly all positive classes (0.64 and 0.68, respectively, for recall, and 0.67 and 0.71, for f–measure), although having good precision measures. The consensus resulting from negotiation was better than the individual results obtained by these agents, having identified correctly all positive classes (recall equals 1 for all groups of ontologies). The semantic agent had better performance than lexical and structural agents (recall equals 1 and f–measure equals 0.78), and it produces similar results when compared with the negotiation consensus. For ontologies which are lexically and structurally simple (e.g., "Company profiles"), all agents produce equivalent results.

**Table 3.** Matcher agents and consensus results.

| | Consensus | | | Lexical | | | Semantic | | | Structural | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | P | R | F | P | R | F | P | R | F | P | R | F |
| Google-Yahoo dir. (54) | 0.33 | 1.0 | 0.49 | 0.50 | 0.25 | 0.33 | 0.28 | 1.0 | 0.43 | 1.0 | 0.50 | 0.66 |
| Product schemas (30) | 0.80 | 1.0 | 0.88 | 0.40 | 0.50 | 0.44 | 0.80 | 1.0 | 0.88 | 0.60 | 0.75 | 0.66 |
| Course catalogs (48) | 0.66 | 1.0 | 0.79 | 1.0 | 0.83 | 0.90 | 0.66 | 1.0 | 0.79 | 0.60 | 0.50 | 0.54 |
| Company profiles (9) | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Average | 0.69 | 1.0 | 0.79 | 0.72 | 0.64 | 0.67 | 0.68 | 1.0 | 0.78 | 0.80 | 0.68 | 0.71 |

The similar results between semantic agent and negotiation consensus occurs because the labels mapped by experts have strong semantic correspondence, identified as "mappings with certainty" by the semantic agent. In these cases, the structural agent returned "mappings with uncertainty", while the lexical agent returned "not mappings with certainty" (e.g., the correct mapping between "/Arts/Arts_History" and "/Architecture/History" terms). Then, the semantic agent decides the final category. However, for the "Google and Yahoo" ontologies, which have greater number of terms (54) when compared with the other groups of ontologies, the consensus returned better precision (0.33) than semantic agent (0.28). As a concluding result, the consensus had better behavior than lexical, semantic and structural individual agents, with f–measure value equals 0.79 against 0.67, 0.78 and 0.71, respectively.

We also identified cases where conflicts occur, which are not resolved by our model and the semantic agent is not sufficient to identify them. Considering the terms "Music/History" and "Architecture/History" ("Google and Yahoo" ontologies), the semantic and lexical agents returned a "mapping with certainty", differently of the structural agent. However, this is not a correct mapping. As will be commented in section 7, we are working on argument-based negotiation, in order to solve this kind of conflict. An argument for accepting the mapping may be that the terms are synonymous and an argument against may be that some of their super-concepts are not mapped.

Third, we compared our negotiation model with three state of the art matching systems: Cupid[19], COMA[7], and S-Match[14]. The comparative results among these three systems are available in [14]. We utilized these test results as criteria to evaluate our proposal, but the details of these tests (implementations, time of run, processor, etc) are not available. Following, we describe each system.

The Cupid algorithm is based on linguistic and structural approaches. In a first phase, called linguistic matching, it matches individual schema elements based on their names, data types, domains, etc. A thesaurus is used to help match names by identifying short-forms (for instance, Qty for Quantity), acronyms, and synonyms. The result is a linguistic similarity coefficient, lsim, between each pair of elements. The second phase is the structural matching of schema elements based on the similarity of their contexts or vicinities. The structural match depends in part on linguistic matches calculated in phase one and the result is a structural similarity coefficient, ssim, for each pair of elements. The weighted similarity (wsim) is a mean of lsim and ssim: wsim = wstruct $\times$ ssim + (1$-$wstruct) $\times$ lsim, where the constant wstruct is in the range 0 to 1.

The 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 value between 0 (strong dissimilarity) and 1 (strong similarity) 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.

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

Our proposal uses different techniques for composite mapping approaches from these previous work.

Our comparative results consider the mappings between attributes of the ontologies in order to compute the precision and recall measures. Then, we have added to our ontologies such attributes, which are viewed as specific sub-classes by our agents. Table 4 shows the comparative results. Considering the attributes of the ontologies, the number of terms to be compared is 160 (i.e., 10 terms in the first ontology with 16 terms in the second ontology).

**Table 4.** Comparative mapping results – matching systems and negotiation model.

| | Consensus | | | Cupid | | | COMA | | | S-Match | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | 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 |

As shown in Table 4, our model returned better precision than Cupid and COMA, and similar precision when compared to the S-Match, having returned as "mapping with certainty" only the correct expert mappings (precision equals to 1). When comparing the F-measure values, our model had similar result than COMA and S-Match and better result than Cupid.

In order to obtain better results than our negotiation model, we propose extend the model using the argumentation formalism. In the following sections, we first introduce the argumentation formalism. Next, we present our novel argumentation model and its evaluation. using it.

## 6 Argumentation Framework

Our argumentation model is based on the Value-based Argumentation Frameworks (VAF)[3], a development of the classical argument system of Dung [9]. First, we present the Dung's framework, upon which a VAF rely. Next, we present a VAF and our extended framework.

### 6.1 Classical argumentation framework

Dung [9]defines an argumentation framework as follows.

**Definition 2.1.1** An Argumentation Framework is a pair $AF = (AR, attacks)$, where $AR$ is a set of arguments and *attacks* is a binary relation on AR, i.e., *attacks* $\subseteq$ AR $\times$ AR. An *attack*(A,B) means that the argument A attacks the argument B. A set of arguments $S$ attacks an argument B if B is attacked by an argument in $S$.

The key question about the framework is whether a given argument $A$, $A \in AR$, should be accepted. One reasonable view is that an argument should be accepted only if every attack on it is rebutted by an accepted argument [3]. This notion produces the following definitions:

**Definition 2.1.2** An argument $A \in AR$ is *acceptable* with respect to set arguments $S(acceptable(A,S))$, if $(\forall\, x)(x \in AR)$ *&* $(attacks(x,A)) \longrightarrow (\exists\, y)(y \in S)$*&*$attacks(y,x)$

**Definition 2.1.3** A set $S$ of arguments is *conflict-free* if $\neg(\exists\, x)(\exists\, y)((x \in S)$*&*$(y \in S)$*&*$attacks(x,y))$

**Definition 2.1.4** A conflict-free set of arguments $S$ is *admissible* if $(\forall x)(x \in S) \longrightarrow acceptable(x,S)$

**Definition 2.1.5** A set of arguments $S$ in an argumentation framework $AF$ is a *preferred extension* if it is a maximal (with respect to set inclusion) admissible set of $AR$.

A *preferred extension* represent a consistent position within $AF$, which can defend itself against all attacks and which cannot be further extended without introducing a conflict.

The purpose in extending the AF is to allow to distinguish between one argument attacking another, and that attack succeeding, so that the attacked argument is defeated.

### 6.2 Value-based argumentation framework

In Dung's frameworks, attacks always succeed. However, in many domains, including the one under consideration, arguments lack this coercive force: they provide reasons which may be more or less persuasive [21]. Moreover, their persuasiveness may vary according to their audience. The VAF is able to distinguish attacks from successful attacks, those which defeat the attacked argument. It allows relate strengths of arguments to their motivations and accommodate different audiences with different interests and preferences.

**Definition 2.2.1** A Value-based Argumentation Framework (VAF) is a 5-tuple $VAF = (AR, attacks, V, val, P)$ where $(AR, attacks)$ is an argumentation framework, $V$ is a nonempty set of values, $val$ is a function which maps from elements of $AR$ to elements of $V$ and $P$ is a set of possible audiences. For each $A \in AF$, $val(A) \in V$.

**Definition 2.2.2** An audience-specific value based argumentation framework (AVAF) is a 5-tuple $VAF_a = (AR,\ attacks, V, val, Valpref_a)$ where $AR$, $attacks, V$ *and* $val$ are as for the VAF, $a$ is an audience and $Valpref_a$ is a preference relation (transitive, irreflexive and asymmetric) $Valpref_a \subseteq V \times V$, reflecting the value preferences of audience $a$. $Valpref(v_1,v_2)$ means $v_1$ is preferred to $v_2$.

**Definition 2.2.3** An argument $A \in AF$ defeats$_a$ (or *successful attacks*) an argument $B \in AF$ for audience $a$ if and only if both *attacks(A,B)* and not *valpref(val(B), val(A))*.

An attack succeeds if both arguments relate to the same value, or if no preference value between the values has been defined.

**Definition 2.2.4** An argument $A \in AR$ is *acceptable* to audience $a$ (*acceptable*$_a$) with respect to set of arguments $S$, *acceptable*$_a$*(A,S))* if $(\forall\ x)\ ((x \in AR\ \&\ defeats_a\ (x,A)) \longrightarrow (\exists y)((y \in S) \&\ defeats_a(y,x)))$.

**Definition 2.2.5** A set $S$ of arguments is *conflict-free* for audience $a$ if $(\forall\ x)(\forall\ y)((x \in S\ \&\ y \in S) \longrightarrow (\neg attacks(x,y) \vee valpref(val(y),val(x)) \in valpref_a))$.

**Definition 2.2.6** A *conflict-free* for audience $a$ set of argument $S$ is *admissible* for an audience $a$ if $(\forall x)(x \in S \longrightarrow acceptable_a(s,S))$.

**Definition 2.2.7** A set of argument $S$ in the VAF is a *preferred extension* for audience $a$ *(preferred$_a$)* if it is a maximal (with respect to set inclusion) *admissible* for audience $a$ of AR.

In order to determine the preferred extension with respect to a value ordering promoted by distinct audiences, [3] introduces the notion of *objective* and *subjective* acceptance.

**Definition 2.2.8** An argument $x \in AR$ is *subjectively* acceptable if and only if $x$ appears in the preferred extension for some specific audiences but not all. An argument $x \in AR$ is *objectively* acceptable if and only if, $x$ appears in the preferred extension for every specific audience.
An argument which is neither objectively nor subjectively acceptable is said to be *indefensible*.

### 6.3 An extended value-based argumentation framework

We extend the VAF in order to represent arguments with confidence degrees. Two elements have been added to VAF: a set with confidence degrees and a function which maps from confidence degrees to arguments. The confidence value represents the confidence that a specific agent has in some argument. We assumed that the confidence degrees compose a second axis which is necessary to represent a problem domain, such as the ontology mapping.

**Definition 2.3.1** An Extended Value-based Argumentation Framework (E-VAF) is a 7-tuple $E\text{-}VAF = (AR, attacks, V, val, P, C, valC)$ where $(AR, attacks, V, val, P)$ is a value-based argumentation framework, $C$ is a nonempty set of values representing the confidence degrees, *valC* is a function which maps from elements of $AR$ to elements of $C$. $valC \subseteq C \times C$ and $valprefC(c_1,c_2)$ means $c_1$ is preferred to $c_2$.

**Definition 2.3.2** An argument $A \in AF$ defeats$_a$ (or *successful attacks*) an argument $B \in AF$ for audience $a$ if and only if *attacks(A,B)* and (*valprefC(valC(A),valC(B))* or ($\neg$ *valpref(val(B),val(A))* and $\neg$ *valprefC(valC(B), valC(A))*))).

An attack succeeds if (a) the confidence degree of the attacking argument is greater than the confidence degree 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 confidence degree of the argument being attacked is not greater than the attacking argument.

**Definition 2.3.4** A set $S$ of arguments is *conflict-free* for audience $a$ if *$(\forall A)(\forall B)$ $((A \in S$ & $B \in S) \longrightarrow (\neg$attacks(A, B) $\vee$ ($\neg$valprefC(valC(A),valC(B)) and (valpref(val(B), val(A)) $\vee$ valprefC(valC(B),valC(A))))).*

# 7 E-VAF for Composite Ontology Mapping

In our model, dedicated agents encapsulate different mapping approaches which represent different audiences in an E-VAF, i.e, the agents' preferences are based on specific approach used by the agent. In this paper we will consider three argumentive audiences: lexical (L), semantic (S), and structural (E) (i.e. $P = \{L, S, E\}$, where $P \in$ E-AVF). We point out that our model is extensible to others audiences.

First, we present the re-organization of the agents society and next we detail the argumentation process.

## 7.1 Organization of the agents society

We use the Moise+ model to describe our novel argumentation model (Figure 6). In this society, only the matcher role is identified, which is responsible for giving an output between two ontology mappings (i.e., encapsulate the mapping algorithms). One matcher could assume the lexical, semantic or structural role. Differently from our negotiation model, there is no role responsible for mediating the argumentation process. The mediator role has been eliminated.

The possible relation between the agents is the communication, where the agents playing a source role are allowed to communicate with agents playing the destination role. This kind of relation is present through the communication among the three matcher agents within an agent society.

## 7.2 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 all possible correspondences between terms of two ontologies. A mapping can

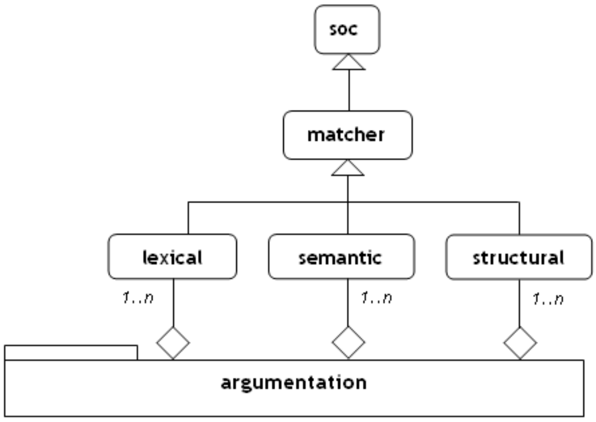


**Fig. 6.** Organizational model.

be described as a 3-tuple $m$ = $(t_1,t_2,R)$, where $t_1$ corresponds to a term in the ontology 1, $t_2$ corresponds to a term in the ontology 2, and $R$ is the mapping relation resulting from the mapping for these two terms. The lexical and semantic agents are able to return *equivalence* values to $R$, while the structural agents returns *sub-class* or *super-class* values to $R$.

Each mapping $m$ is represented as a argument. Now, we can define arguments as follows:

**Definition 4.1** An *argument* $\in AF$ is a 4-tuple $x$ = $(m,a,c,h)$, where $m$ is a mapping; $a \in P$ is the agent's audience generating that argument; $c \in C$ is the confidence degree associated to that mapping; $h$ is one of {-,+} depending on whether the argument is that $m$ does or does not hold.

The confidence degree is defined by the agent when applying the specific mapping approach. Here, we assumed $C$ = {certainty, uncertainty}, where $C \in$ E-VAF.

Table 5 shows the possible values to $h$ and $c$, according to the agent's audiences. The agents generate theirs arguments based on rules from Table 5.

**Table 5.** $h$ and $c$ to audiences.

| | | Audiences | |
|---|---|---|---|
| $h$ | $c$ | Lexical | Semantic |
| + | certainty | 1 | synonym |
| + | uncertainty | 1 > r > t | related |
| - | certainty | 0 < r <= t | |
| - | uncertainty | 0 | unknown |

**Lexical agent.** The output of lexical agents ($r$) is a value from the interval [0,1], where 1 indicates high similarity between two terms. This way, if the output is 1, the lexical agent generates an argument $x$ = (m,L,*certainty*,+), where $m$ = ($t_1$,$t_2$,*equivalence*).

If the output is 0, the agent generates an argument $x$ = (m,L,*certainty*,-), where $m$ = ($t_1$,$t_2$, *equivalence*). A threshold ($t$) is used to classify the output in uncertain categories. The threshold value can be specified by the user.

**Semantic agent.** The semantic agents consider semantic relations between terms, such as synonym, antonym, holonym, meronym, hyponym, and hypernym (i.e., such as in WordNet database). When the terms being mapped are synonymous, the agent generates an argument $x$ = (m,S,*certainty*,+), where $m$= ($t_1$,$t_2$, *equivalence*).

The terms related by holonym, meronym, hyponym, or hypernym are considered related and an argument $x$ = (m,S, *uncertainty*,+) is generated, where $m$ =($t_1$,$t_2$, *equivalence*); when the terms can not be related by the WordNet (the terms are unknown for the WordNet database), an argument $x$ = (m,L,*uncertainty*,-), where $m$ = ($t_1$,$t_2$,*equivalence*), is then generated.

**Structural agent.** The structural agents consider the super-classes (or sub-classes) intuition to verify if the terms can be mapped. First, it is verified if the super-classes are lexically similar. If not, the semantic similarity is used. If the super-classes are lexically or semantically similar, the terms are equivalent to each other. The argument will be generated according to the lexical or semantic comparison.

For instance, if super-classes are not lexically similar, but the terms are considered synonymous, an argument $x$ = (m,E,*certainty*,+), where $m$ = ($t_1$,$t_2$, *super-class*), is generated.

### 7.3 Preferred extension generation

After generating their set of arguments, the agents change with each other their arguments. Following a well-defined protocol, an agent asks the others about theirs arguments. The other agents then, send their arguments to the first agent. An *ack* sign is then sent to requesting agents, in order to indicate that the arguments have been correctly received. Otherwise, an *error* sign is sent. Figure 7 shows an AUML interaction diagram with the messages exchanged between the agents during the argumentation process.

When all agents have received the set of argument 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,*certainty*,+) have as an *attack* an argument $y$ = ($m_2$,E, *certainty*,-), where $m_1$ and $m_2$ have the same terms in the ontologies. The argument $y$ also represents an *attack* to the argument $x$.

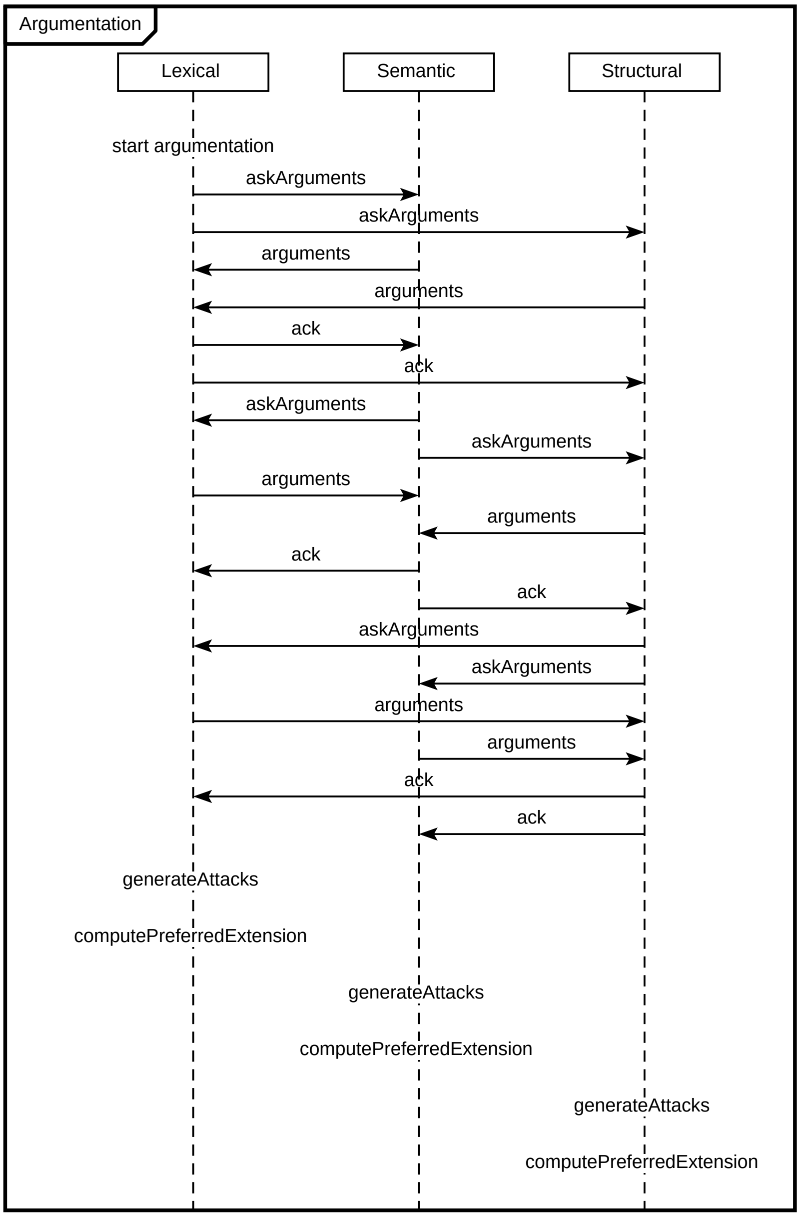


**Fig. 7.** AUML interaction diagram.

As an example, consider the mapping between the terms “Reference/ Dissertation” and “Citation/Thesis” and the lexical and structural agents. The lexical agent generates an argument $x$ = (m,L,*uncertainty*,-), where $m$ = (dissertation,thesis,*equivalence*); and the structural agent generates an argument $y$ = (m,E,*certainty*,+), where $m$ = (dissertation,thesis, *super-class*). For both lexical and structural audiences, the set of arguments is $AR$= {x,y} and the *attacks* = {(x,y),(y,x)}. However, the relations of *successful attacks* will be defined according to specific audience (see *Definition 2.3.2*), as it is commented below.

When the set of arguments and attacks have been produced, it is necessary for the agents to consider which of them they should accept. To do this, the agents compute their preferred extension, according to the audiences and confidence degrees. 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 structural(E) audiences, where L $\succ$ E and E $\succ$ L, respectively. For the lexical audience, the argument $y$ successful attacks the argument $x$, while the argument $x$ does not successful attack the argument $y$ for the structural audience. Then, the preferred extension of both lexical and structural agents is composed by the argument $y$, which can be seen as globally *objectively* acceptable. The mapping between the terms “Reference/ Dissertation” and “Citation/Thesis”, indicated by $y$ is correct.

## 8 Experiments using the E-VAF

Let us consider that three agents need to obtain a consensus about mappings that link corresponding class names in two different ontologies.

First, we considered part of the ontology of Google and Yahoo web directories[10], and the argumentation model output have been compared with manual matches[11] (expert mappings).

We considered lexical (L), semantic (S), and structural (E) audiences in order to verify the behavior of our argumentation model. These agents were implemented in Java, and the experiments ran on Pentium(R) 4, UCP 3.20GHz, 512MB. The argumentation model, however, was not fully implemented. In order to have its practical evaluation, the output of the agents were used as input for a manual simulation of the argumentation protocol.

The threshold used to classify the matcher agents output was 0.6. We have selected three possible mappings between terms of the ontologies: “Music/History” and “Architecture/History”, “Art/ArtHistory” and “ArtHumanity/ArtHistory”, and “Art” and “ArtHumanity”. Table 6 shows arguments and attacks (counter-arguments) generated for each audience. The mappings between these terms

[10] http://dit.unitn.it/ãccord/Experimentaldesign.html (Test 3)
[11] http://dit.unitn.it/ accord/Experimentaldesign.html

have been selected because they were identified as conflicting cases when using our negotiation model.

**Table 6.** Arguments and attacks.

| ID | Argument | Attacks |
|---|---|---|
| 1 | (history,history,*equivalence*,L,*certainty*,+) | 3 |
| 2 | (history,history,*equivalence*,S,*certainty*,+) | 3 |
| 3 | (history,history,*super-class*,E,*certainty*,-) | 1,2 |
| 4 | (art-history,art-history,*equivalence*,L,*certainty*,+) | - |
| 5 | (art-history,art-history,*equivalence*,S,*certainty*,+) | - |
| 6 | (art-history,art-history,*super-class*,E,*certainty*,+) | - |
| 7 | (art,art-humanity,L,*equivalence*,*uncertainty*,-) | 8,9 |
| 8 | (art,art-humanity,S,*equivalence*,*certainty*,+) | 7 |
| 9 | (art,art-humanity,E,*super-class*,*uncertainty*,+) | 7 |

For the mapping between the terms "Music/History" and "Architecture/ History", each agent has as arguments $AR = \{1,2,3\}$ and as relations of attack $attacks = \{(3,1), (3,2), (1,3), (2,3)\}$. These sets are generated by each agent, after receiving the arguments of the other agents. After, the arguments that defeat each other are computed. For the lexical audience, where L $\succ$ S and L $\succ$ E, there is no arguments that successful attack each other, because all agent have certainty in the mappings. The same occurs for the semantic (S $\succ$ L and S $\succ$ E) and structural (E $\succ$ L and E $\succ$ S) audiences.

Then, the preferred extensions of the agents are composed by the arguments generated by the corresponding agent (i.e, the preferred extension of the lexical agent is $\{1\}$; the preferred extension of the semantic agent is $\{2\}$; and the preferred extension of the structural agent is $\{3\}$). This way, there is no argument globally *objectively* acceptable. We can consider that the mapping between the terms is not possible, what is true according to the manual mapping.

Using our negotiation model, the final mapping between the "Music/History" and "Architecture/ History" terms was incorrect. The semantic and lexical agents returned mappings with certainty, while the structural agent returned a not mapping with certainty. By majority, the mapping with certainty was obtained. This conflict is then resolved by our argumentation model.

For the mapping between the terms "Art/ArtHistory" and "ArtHumanity/ArtHistory", each agent has as arguments $AR = \{4,5,6\}$, but there are not relations of attack. Then, all agents accept the mapping with certainty between these terms. This mapping is considered a correct mapping by the manual mapping.

Finally, for the mapping between the terms "Art" and "ArtHumanity", each agent has as arguments $AR = \{7,8,9\}$ and as relations of attack $attacks = \{(8,7), (9,7), (7,8), (7,9)\}$. For the lexical audience, the argument 8 successful attacks the argument 7. Then, the preferred extension has the argument 8. For the semantic audience, the argument 8 also successful attacks the argument 7, and for audience structural, the arguments 8 and 9 successful attack theirs counter-arguments. Then, the preferred extension of the structural agent is {8,9}. The argument 8 is present in all preferred extension, then it is globally *objectively* acceptable, confirming the mapping indicated by manual mapping.

We have used different agents' output which use distinct mapping algorithms in order to verify the behavior of our model. Our argumentation model has identified correctly the three mappings defined by expert mappings, being two mapping positives ($h$ is $+$) and one negative ($h$ is -).

Second, we compared the argumentation output with the results obtained by a cooperative negotiation model. Table 7 shows the comparative results. Although the negotiation model having obtained better precision than argumentation model, the F-measure of the argumentation model is better than negotiation model. The negotiation model identified 7 true positive mappings and it did not classify correctly 4 true positive mappings. The argumentation model identified 8 true positive, returning 1 false positive mapping not identifying 3 true positives mappings.

**Table 7.** Argumentation vs. negotiation.

| | Argumentation | | | Negotiation | | |
|---|---|---|---|---|---|---|
| Ontology | P | R | F | P | R | F |
| Company profiles (160) | 0.88 | 0.72 | 0.79 | 1 | 0.63 | 0.77 |

Third, we compared our argumentation model with Cupid, COMA, and S-Match systems. We consider the class and the attribute names of the ontologies in the comparison. Table 8 shows the results. Our argumentation model had better F-measures than all others systems.

**Table 8.** Comparative mapping results – argumentation model.

| | Arg | | | Cupid | | | COMA | | | S-Match | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | P | R | F | P | R | F | P | R | F | P | R | F |
| Company profiles (160) | 0.88 | 0.72 | 0.79 | 0.50 | 0.60 | 0.54 | 0.80 | 0.70 | 0.74 | 1.0 | 0.65 | 0.78 |

## 9 Related Work

In the field of ontology negotiation we find distinct proposals. [35] 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 [6], where ontologies are integrated to support the communication among heterogeneous agents.

[1] 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 the argumentation formalism. The links between related concepts are the result of the preferred mappings of each agent, instead of an integrated ontology upon which the agents will be able to communicate for a specific purpose. We do not consider negotiation steps such as the ones presented in [1], namely clarification and explanation. But we consider different mapping methods represented by different audiences selecting by argumentation the best solution for the mapping problem.

[32] 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 [24]. 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 [32], 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. The arguments in each preferred set are considered globally acceptable.

[21] proposes to use an argument framework to deal with arguments that support or oppose candidate correspondences between ontologies. The mapping candidates are provided by a single service. The accepted mappings resulting from argumentation are used to agent communication. Differently from [21], the mappings are obtained by different agents specialized on different mapping algorithms and not only in a single service. In [21], the mappings are assumed to be correct, and we are interested in how to obtain mapping sets by combining different approaches for ontology mapping. Moreover, in [21] it is assumed that arguments being negotiated have the same confidence. We are proposing to associate to each argument a confidence degree. This way, in order to compute the preferred mapping, the audiences and confidence degrees must be considered.

Semantic heterogeneity is an important problem for data bases and more recently it has been raised as one of the key problems to be solved for the development of the semantic web. We can find in the literature different approaches to the problem. The work presented in [29] provides an encoding of the extensible knowledge on commonly found semantic conflicts, providing an automatic way of comparing and manipulating contextual knowledge of different information sources, which is used for semantic transformation across heterogeneous databases. In [33], the MAFRA Toolkit is presented, the tool helps a domain expert to work on ontology mapping tasks. Whereas these previous approaches are concerned with the specification of semantic conflicts that arise between different sources, ours is concerned with the particular problem of identifying pairs of corresponding terms in different ontologies. In the future we will see these various approaches in an integrated way.

## 10 Final Remarks

This paper presented the use of cooperative agents for composite ontology mapping. We first presented an extended classification on automated ontology matching and proposed an automatic composite solution for the matching problem based on cooperative approach. Our agents encapsulate different mapping approaches (lexical, semantic and structural) and a consensual result from cooperative negotiation of these agents. This model is fully implemented. We compared our results with expert mappings, for four ontologies in different domains. The negotiation result was better than lexical and structural agents and it returned better F-measure value than the semantic agent. When comparing our model with other state of the art matching systems, our model obtained better F-measure than Cupid and COMA and similar results if compared with the S-Match system.

Next, we proposed an extension of our negotiation model, which is based on argumentation formalism. With this we were able to give a formal presentation of our composite mapping approach. Our proposal extends the Value-based Argumentation Framework, in order to represent arguments with confidence degrees. We assumed that the confidence degrees compose a second axis which is necessary to represent a problem domain, such as the ontology mapping. We initially evaluated the argumentation model considering the mapping identified as conflicting cases when using the negotiation model. This model has obtained satisfactory results for the conflicting cases. We also compared the argumentation model with the Cupid, COMA, and S-Match. Our model obtained better F-measure values than these systems. The contribution of the argumentation model, which is the only one that is not implemented resides in the formal presentation of the problem, which was given with its practical evaluation.

As future work we plan to improve the semantic and structural approaches. We also intend to develop further tests considering also agents using constraint-based approaches; and use the ontology's ap- plication context in our matching approach. Another goal is to evaluate our proposal against systems based on machine learning techniques, such as the system proposed by [8]. Finally, we plan to use the mapping result as input to an ontology merging process in the question answering domain.

## References


1. S. Bailin and W. Truszkowski. Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*, 17(1):7–19, 2002.
2. M. Beer, M. d'Inverno, M. Luck, N. Jennings, C. Preist, and M. Schroeder. Negotiation in multi-agent systems. In *Workshop of the UK Special Interest Group on Multi-Agent Systems*, 1998.
3. T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.
4. J. Bigham and L. Du. Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network. In *Proceedings of the Second*

*International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 568–575. ACM Press, 2003.
5. R. Dieng and S. Hug. Comparison of personal ontologies represented through conceptual graphs. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, page 341, 1998.
6. 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 V International Conference on Autonomous Agents and Multi-Agent Systems*, pages 899–906, 2006.
7. 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 (VLDB)*, 2002.
8. A. Doan, J. Madhaven, R. Dhamankar, P. Domingos, and A. Helevy. Learning to match ontologies on the semantic web. *VLDB Journal (Special Issue on the Semantic Web)*, 2003.
9. 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.
10. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *Proceedings of the European Semantic Web Symposium*, pages 76–91, 2004.
11. J. Euzenat, T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng-Kuntz, M. Ehrig, M. Hauswirth, M. Jarrar, R. Lara, D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris, S. Van Acker, and I. Zaihrayeu. State of the art on ontology alignment. Technical report, 2004.
12. N. Gatti and F. Amigoni. A cooperative negotiation protocol for physiological model combination. In *Proceedings of the Third Internation Joint Conference on Automomous Agents and Multi-Agent Systems*, pages 655–662, 2004.
13. F. Giunchiglia and P. Shvaiko. Semantic matching. *Knowledge Engineering Review*, 18(3):265–280, 2004.
14. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: An algorithm and an implementation of semantic matching. In *First European Semantic Web Symposium*, 2004.
15. S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, and R. Evans. Software agents: A review. Technical report, Trinity College, 1997.
16. E. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings of the First International conference on language resources and evaluation*, 1998.
17. J. Hubner. *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Escola Politécnica da Universidades de São Paulo, Departamento de Engenharia da Computação e Sistemas Digitais, 2003.
18. J. Hubner, J. Sichman, and O. Boisser. A model for structural, functional, and deontic specification of organizations in multiagent systems. In *Advances in Artificial Intelligence*, 2002.
19. P. B. J. Madhavan and E. Rahm. Generic schema matching with cupid. In *Proceedings of the Very Large Data Bases Conference (VLDB)*, page 49, 2001.
20. Y. Kalfoglou and W. M. Schorlemmer. Ontology mapping: The state of the art. In *Semantic Interoperability and Integration*, 2005.
21. 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).*

22. S. Lander and V. Lesser. Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, January 1993.
23. I. Levenshtein. Binary codes capable of correcting deletions, insertions an reversals. In *Cybernetics and Control Theory*, 1966.
24. 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.
25. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management*, pages 251–263, 2002.
26. M. Mailler, V. Lesser, and B. Horling. Cooperative negotiation for soft real-time distributed resource allocation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 576–583. ACM Press, 2003.
27. N. Noy and M. Musen. Anchor-prompt: using non-local context for semantic matching. In *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–70, 2001.
28. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB*, 10:334–350, 2001.
29. S. Ram and J. Park. Semantic conflict resolution ontology (scrol): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):189–202, 2004.
30. P. Shvaiko. A classification of schema-based matching approaches. Technical report, Informatica e Telecomunicazioni, University of Trento, 2004.
31. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. Technical report, Informatica e Telecomunicazioni, University of Trento, 2004.
32. N. Silva, P. Maio, and J. Rocha. An approach to ontology mapping negotiation. In *Proceedings of the K-CAP Workshop on Integrating Ontologies*.
33. N. Silva and J. Rocha. Semantic web complex ontology mapping. *wi*, 0:82, 2003.
34. T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
35. 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.
36. C. Trojahn, M. Moraes, P. Quaresma, and R. Vieira. A negotiation model for ontology mapping. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2006.
37. M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons, 2002.
38. X. Zhang, V. Lesser, and R. Podorozhny. Multi-dimensional, multistep negoriation for task allocation in a cooperative system. *Autonomous Agents and Multi-Agent Systems*, 10:5–40, 2005.