Enhancing Interoperability: Ontology – Mapping in an Electronic Institution

Henrique Lopes Cardoso, Daniel Dinis Teixeira, and Eugénio Oliveira

LIACC, DEI / Faculdade de Engenharia, Universidade do Porto
R. Dr. Roberto Frias, 4200-465 Porto, Portugal
`hlc@fe.up.pt, daniel.teixeira@fe.up.pt, eco@fe.up.pt`

Abstract. The automation of B2B processes requires a high level of interoperability between potentially disparate systems. We model such systems using software agents (representing enterprises), which interact using specific protocols. When considering open environments, interoperability problems are even more challenging. Addressing business automation as a task that intends to align businesses through a tight integration of processes may not be desirable, because business relationships may be temporary and dynamic. Furthermore, openness implies heterogeneity of technologies, processes, and even domain ontologies. After discussing these issues, this paper presents, in the context of an Electronic Institution, an ontology-mapping service that enables the automation of negotiation protocols when agents may use different ontologies to represent their domain knowledge. The ontology-mapping service employs two approaches used for lexical and semantic similarity, namely *N-Grams* and *WordNet*, and poses few requirements on the ontologies' representation format. Examples are provided that illustrate the integration of ontology-mapping with automated negotiation.

Keywords: Automated negotiation, Open environment, Heterogeneity problem, Ontology-mapping.

1 Introduction

Technological support to the creation of B2B relationships is arising in many forms. The most ambitious ones intend to automate (part of) the process of creation and execution of contracts, mainly through multi-agent system (MAS) approaches.

The agent technology roadmap [1] identifies as key problem areas the development of infrastructures for open agent communities, as well as the need for trust and reputation mechanisms. Electronic institutions, together with ontologies and related services, address the needed infrastructures. Norms, electronic contracts and their enforcement are pointed out as means to achieve trust in open environments.

A keyword in these recommendations is *open*. In open environments interoperability problems are exacerbated, posing further challenges on the efforts to solve them. In fact, addressing an open environment implies that one intends to accommodate a wider range of agent architectures, technologies, or representation formalisms. Appropriate tools are needed in order to assist interoperation among such disparate systems.

K. Fischer et al. (Eds.): ATOP 2005 and ATOP 2008, LNBIP 25, pp. 47–62, 2009.

We will start by briefly discussing the implication of openness in business automation approaches using MAS, according to our view. We will then concentrate on the issue of ontologies in open environments, giving a motivation for the ontology-mapping approach that we describe throughout the rest of the paper.

1.1 Open Agent Communities for Business Scenarios

Today's business environments are characterized by a high degree of specialization and frequent market changes. The *Virtual Enterprise* (VE) concept is, as a consequence, a major trend in enterprise interoperability. A VE has been defined as "a temporary consortium of autonomous, diverse and possibly geographically dispersed organizations that pool their resources to meet short-term objectives and exploit fast-changing market trends" [2]. In outsourcing or supply chain configurations, a tight integration of partners using information technology approaches (focused on managing inter-organizational workflows with, e.g., Web-services, SoA, and service composition with BPEL) provides a fine-grained engagement between parties that leads to middle- or long-term relationships. However, flexible and dynamic relationships are the trend in a very competitive market. Because of this, we address interoperability in open business scenarios at a legal/contractual level [3]. Contracts formalize partners' commitments in a way that allows for their monitoring and enforcement.

In "breeding environments" [4] potential partners are already acquaintances and typically have previous common business experiences. This facilitates the construction of business agreements, as parties can rely on relational contracts [5], which specify continuous relationships that are naturally self-enforceable. However, in open environments potential partners may not be acquaintances, meaning that a business engagement may comprise entities that have never worked together in the past. In this more open setting formal contracts and their enforcement are a means to ensure trust.

Looking from a MAS perspective, while agent theory describes agents as autonomous self-interested entities, preferably interacting in open environments, an important issue arises when attempting to apply agents in real world scenarios: how to ensure cooperative outcomes in scenarios populated with self-interested agents? A possible answer to this problem is to regulate the environment, providing incentives for cooperative behavior through normative constraints [6].

Taking the aforementioned roadmap [1] into account, we have been developing an Electronic Institution (EI) platform motivated by the need to develop services that assist the coordination efforts of agents which, representing different real-world entities, interact with the aim of establishing business relationships. We therefore aim at agent-based B2B contracting, focusing on process automation. Some of the services we developed are depicted in Figure 1. The negotiation and establishment of electronic contracts are important in business interactions among companies that rely on running their businesses electronically.

The establishment of contractual agreements is supported with negotiation mediation, based on appropriate negotiation protocols and contract templates, defined using an institutional ontology. The validation and registration of contracts allows for their

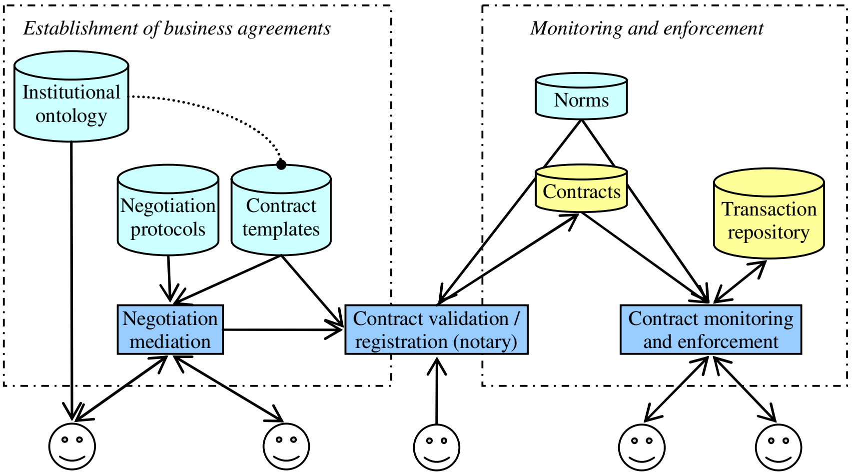


Fig. 1. Some services in an Electronic Institution

"legal" existence. Contracts are created as a result of a successful negotiation. However, we do not assume that agents will always have their negotiations mediated within the EI. As such, agents may opt to use institutional services for compliance checking only. An enforceable normative environment is established by rendering a contract monitoring and enforcement service, which registers transactions and verifies norm applicability, as well as the fulfillment of signed contracts.

Although aiming to address open environments, we do have a set of assumptions regarding agents' interoperability. First of all, agents must be able to 'speak' a common language (ACL in the agents' world [7]). We also assume that there is a common understanding on domain-independent business vocabulary, concerning terms such as 'proposal', 'price', 'delivery' or 'payment'.

We have also assumed in the past that agents have common domain ontologies, an issue that we have later addressed by adding ontology-based services to our infrastructure. This paper describes how such services have been integrated with the contract negotiation phase. Before we describe our approach, in the next subsection we will identify this problem more clearly.

1.2 Ontologies in Open Environments

An intrinsic problem that must be dealt with when approaching open systems is that each of a set of heterogeneous entities may potentially use a different domain ontology. There may be syntactic or semantic discrepancies in these ontologies: the same information may be stored in different representation formats, diverse terminologies for the same concepts may exist, or even the same terminology may be used for distinct concepts. This heterogeneity is a critical impediment to efficient business information exchange and to the automation of B2B processes.

The most simplistic way of solving this problem (often called the *interoperability* or *heterogeneity problem* [8, 9]) would be to define either a common ontology (used by all) or a shared one (in terms of which everyone can communicate) which could be understood by all agents participating in business interactions (inside a "breeding environment" [4] common domain ontologies might prevail). However, open environments (where a central design is neither possible nor desirable) populated with heterogeneous agents make the common ontology case unfeasible. Each agent will typically use a different ontology, and enterprises will not consider converting all the content of their ontologies if the target ontology is less expressive or not considered as a *de facto* standard.

The Foundation for Physical Intelligent Agents (FIPA) [10] has analyzed the interoperability problem in heterogeneous Multi-Agent Systems (MAS) and has proposed an Ontology Agent (OA) for MAS platforms [11]. Among other responsibilities, the OA may provide the translation service of expressions between different ontologies or different content languages by itself, possibly as a wrapper to an ontology server. In this paper we present an implementation of such a service, embedded in an Electronic Institution. The ontology-mapping service is aligned with a negotiation mediation service, allowing negotiation to take place between entities using different domain ontologies.

The rest of the paper is structured as follows. Section 2 contextualizes the usage of an ontology-mapping service in agent-based automated contracting. Section 3 details the service itself. In Section 4 examples that exploit the service are provided. Section 5 concludes.

2 Interoperability in Agent-Based Automated Contracting

There is a strong research effort towards the automation of B2B contracting processes. In particular, multi-agent systems technology is being used to establish business contracts by automatically negotiating agreements.

An Electronic Institution (EI) is a software platform that aims at (i) supporting agent interaction as a coordination framework, making the establishment of business agreements more efficient; and at (ii) providing a level of trust by offering an enforceable normative environment [12]. The ontology service described in this paper is essential to serve the first of these aims. Particularly when addressing an open environment, where a central design is not possible, agents representing different enterprises (henceforth *enterprise agents*) may use different domain ontologies, which have to be matched in order to make the (automated) establishment of agreements possible.

The EI will offer a set of services related to contract establishment and execution. A major service concerns negotiation mediation, through which an enterprise agent may automatically find and negotiate with potential partners. Negotiation is typically based on appropriate negotiation protocols and contract templates. The beginning of the negotiation mediation process is where ontology services come into play.

In open environments, different domain-dependent vocabulary may be used by different business entities. Ontology services are important to allow for negotiation to take place. Figure 2 illustrates these concepts.

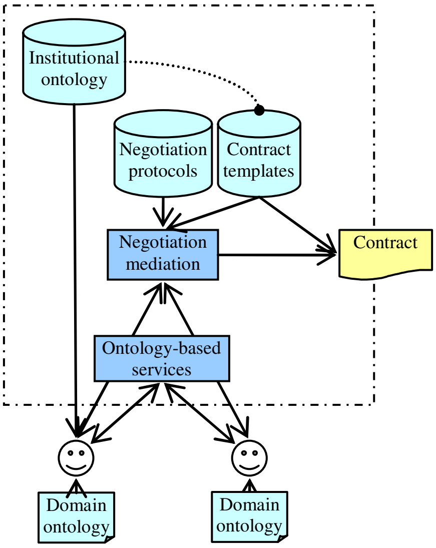


Fig. 2. The interplay between negotiation mediation and ontology services

The negotiation mediation service acts as a mediator between an enterprise agent and a set of potential partners (other enterprise agents). Each of these has a set of competences based on certain classes of components that it is able to supply. When asked for, each enterprise agent can negotiate the supply of a component of a certain class, if that class is in his competence list. Figure 3 illustrates the start of the negotiation process when there is no ontology service for solving the heterogeneity problem: enterprise agents on the right side may be prevented from participating in the negotiation because of an ontology mismatch.

The ontology-mapping service can be used when some enterprise agent does not understand the content of a CFP message (i.e. the component class under negotiation). The agent may recur to the service in order to find out if he supplies components of a class matching the one asked for. Figure 4 illustrates this process. In this case an enterprise agent on the right hand side is able to participate in the negotiation thanks to the ontology-mapping service, which gave him a mapping between the asked component class and one described in his ontology. In order to reduce the potentially large space of mapping attempts that the ontology-mapping service has to do, some heuristics may be used to preselect only those classes that have a potential to successfully map with the target class.

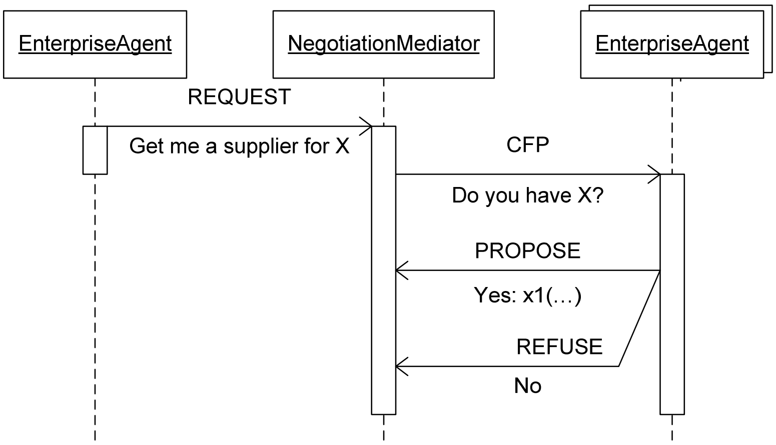


Fig. 3. Negotiation-mediation without an ontology-mapping service

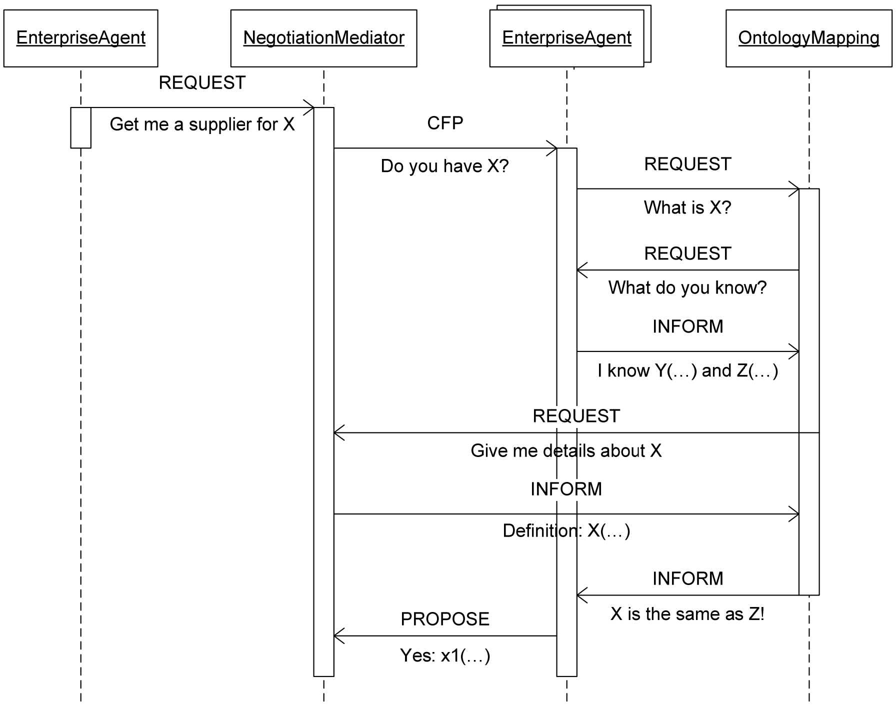


Fig. 4. Negotiation-mediation with an ontology-mapping service

This kind of service is mostly important if we consider that the negotiation process is to be automated through the use of enterprise (software) agents. In the next section we detail the workings of the ontology-mapping service.

3 Ontology-Mapping Service

Our background scenario is based on a set of enterprise agents requesting or supplying certain classes of components, for which they use a negotiation mediation service. Despite their potential interest in the same components, it is not guaranteed that they use the same names to define them. Suppose that a customer wants an 'alarm' and a supplier has exactly the component that this customer is looking for; however, in the supplier's ontology the component is known as a 'siren'. An automated negotiation process will fail if this ontology mismatch is not dealt with. Our approach is based on a service whose aim is to make a mapping between concepts defined in two different ontologies.

This section describes how the mapping process takes place. This process is based on the principle that if two different ontologies represent the same domain, then there is a high probability that the described concepts have a similar syntax and share similar attributes [8]. We will start by describing the minimum set of assumptions that enable the usage of the ontology-mapping service.

3.1 Assumptions on Ontology Representation

Even in open environments, a minimum set of conventions is needed to enable the interaction between heterogeneous agents, be it an ACL, negotiation protocols, and so on. In the B2B domain, it is generally assumed that parties have a common understanding on domain-independent business vocabulary. Concepts like proposal, deal or price must be part of a common base ontology. If we want enterprise agents to automatically negotiate contracts, they should also have a common understanding of what a delivery or a payment means.

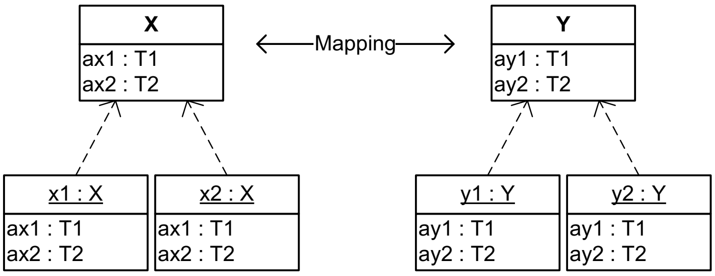


Fig. 5. Classes, attributes and components

In order to render an ontology-mapping service, a minimum set of requirements is also needed regarding the representation of components in different ontologies. Each ontology must be describable in terms of classes and attributes (see Figure 5). Each component is an instance of a class that defines its type. Each class has a name and a set of typed attributes. The mapping-service will be based on matching class names and class attributes.

Since we set the foundations of our approach on lexical and semantic similarity tools (in this case using *WordNet* – an English-based tool), an additional assumption that we make regarding the to-be-matched ontologies is that they are based on the English language.

The following subsection describes how the mapping process takes place.

3.2 Ontology-Mapping

Ontology mapping is the process of finding correspondences between the concepts of two ontologies. If two concepts correspond, then they mean the same thing or closely related things. The mapping process is based on two approaches. The first approach is *N-grams* [13]: an algorithm that takes as input two strings and computes the number of common sub-strings between them. The other approach consists of using *WordNet* [14], which is a free lexical database containing semantic and lexical relations between words. Succinctly, the *N-Grams* algorithm computes a lexical similarity between two words, while *WordNet* computes a semantic one. These two approaches are applied to the names of the classes and also to their attributes, obtaining an overall mapping score between two classes, as explained below.

N-Grams. The *N-Grams* [13] algorithm takes as input two strings and computes the number of common *n-grams* between them. An *n-gram* is a sequence of *n* characters; for each string, the algorithm computes the set of all possible *n-grams* that are in each string. A pre-processing step consists of normalizing both strings: all non-alphanumeric characters are replaced with '_'. The second step is to get the *n-grams* from each string (sub-strings of length *n*). Finally, the algorithm counts the number of *n-grams* of the first string that match an *n-gram* of the second string. The number of matches is used to calculate the outcome of the algorithm – a value of similarity is obtained from the formula:

$$Value = \frac{number\ of\ matches}{number\ of\ n\text{-}grams\ in\ first\ string}$$

The value obtained is within the range [0.0; 1.0]. The algorithm is parameterized with the value of *n* (the size of each *n-gram*). In our approach we chose a value of $n=3$ to produce *3-grams*.

The *N-Grams* similarity approach has been used as an alternative to word-based systems. It has the merit of being robust in misspelling cases, which can be expected to occur in a scenario with multiple ontologies for the same domain.

WordNet. *WordNet* [14] is a lexical database designed for automatic processing that provides an effective combination of traditional lexicographic information and modern computing. *WordNet* contains thousands of words, including nouns, verbs adjectives and adverbs. These words are grouped into sets of cognitive synonyms (*synsets*), each expressing a distinct concept.

Other comparable systems exist [15], but with essentially different purposes, e.g. *CYC* (a general knowledge base and commonsense reasoning engine) or *EDR* (a dictionary with a bilingual English-Japanese emphasis). Comparing to *WordNet*, *CYC* is a more general-purpose system, while *EDR* has a different scope. *WordNet* fits our purposes for being an essentially linguistic knowledge base for English.

Furthermore, we make use of *WordNet::Similarity* [16], a *WordNet*-based Perl module [17] that calculates the similarity or relatedness between a pair of concepts, according to several different measures, such as Resnik, Jiang-Conrath, Leacock-Chodorow, Hirst-St.Onge, Wu-Palmer, among other. Measures of similarity quantify how much two concepts are similar, based on information contained in a hierarchical model. For instance, an 'automobile' can be considered more like a 'boat' than a 'tree', if 'automobile' and 'boat' share 'vehicle' as a common ancestor. Measures of relatedness compare concepts using relations like "has part", "is made of" or "is an attribute" instead of a hierarchical model. For instance a 'wheel' would give a good relatedness with a 'car', since a 'wheel' is an attribute of a 'car' [16].

In previous work [8] we concluded that the most appropriate measure for our scenario is Leacock-Chodorow (LCH), a similarity measure based on path lengths between concepts. LCH finds the shortest path between two concepts and scales that value by the maximum path length in which they occur. We then normalize that value to obtain a result within the range [0.0; 1.0].

In our domain, class and attribute names are not necessarily words appearing in *WordNet*. When names are made from word compositions, it may be the case that they are not part of *WordNet*'s database (e.g. while "photographic equipment" can be found, "vision angle" cannot). In such cases, a pre-processing step dividing the names into words is needed in order to try to find individual word mappings (e.g. the words "vision" and "angle" may be found in *WordNet*), which are then averaged if a mapping is found for them.

Mapping Process. Since we do not know at the beginning if two words have a lexical or semantic similarity, the ideal would be to apply both measures for each pair of words. However, this may not be feasible for performance reasons, because *WordNet::Similarity* [17] has a client/server architecture with socket communications, which introduces a large latency. For this reason, we firstly apply *N-Grams* and only if the result is not satisfactory we make use of *WordNet::Similarity*.

Ontology mapping starts with a list of component classes that can be matched with the requested (i.e. target) component class. Each class in the list will be tested. We choose the best matching class provided that it has a satisfactory value. The following algorithm implements this overall mapping process:

```
1. Let bc be the best matching class and bs its matching score
2. For each class c in the list
     a. Compute c's matching score with target class
     b. Update bc and bs
3. If bs is satisfactory then return bc, otherwise return null
```

The matching score (`2.a` above) between a class and the target class is the average of two values: the similarity score of their names and their attributes. The attribute matching process is done only for attributes of the same type, and is successful only if there is a mapping for every attribute of the target class. The following algorithm shows how the matching score between two classes is calculated:

```
1. Compute the class name similarity score ns
2. Compute the attribute list similarity score as
    a. Let as be the attribute list similarity score
    b. For each attribute at in the target class attribute list
        i. Find the "unmatched" class attribute a with best
           similarity score
        ii. Update as
        iii. Mark a as "matched"
3. Return the average of ns and as
```

Similarity is calculated by first applying *N-Grams* and eventually using *WordNet::Similarity*, as described in the following algorithm:

```
1. Compute the N-Grams similarity score ngs
2. If ngs is satisfactory then return ngs
3. Compute the WordNet::Similarity similarity score wns
4. Return max of ngs and wns
```

As explained above, both `ngs` and `wns` are within the range [0.0; 1.0].

One might think that this approach is rather vulnerable to homonym-like pairs of strings: although syntactically very similar, they could mean entirely different things, and *N-Grams* will give them a high similarity score. While this is the case, it is unlikely that in such a situation the concepts described will have similar attributes, and thus this problem is somewhat contained. In other words, since a matching process involves calculating 1+*n* similarity scores (where *n* is the number of attributes), the homonym problem is unlikely to hinder our approach.

The following section describes an example that shows a scenario with suppliers and customers having different ontologies and where this mapping process is applied.

4 Example

In order to exemplify the usage of the ontology-mapping service and the mapping of several classes, a scenario is described in the following subsections. The scenario includes suppliers and customers interested in components from the domotics domain. It was tested in our Electronic Institution platform developed with the JADE framework [18].

Ontologies were created using the Protégé ontology editor [19] and saved in OWL files. This format allows defining classes of components in an object oriented model, where sub-classes inherit attributes from super-classes. Each enterprise agent instantiates components in an OWL file extended from the ontology definition.

4.1 Scenario

The scenario contains six agents (see Figure 6). Five of them are suppliers (Supply1 to Supply5) and one is a customer (Request1). The customer uses the same ontology (B) as Supply1, Suply2 and Supply3. On the other hand, Supply4 and Supply5 have defined their components based on a different ontology (A). The arrows in Figure 6 show which classes of components are supplied by each of the suppliers. The customer Request1 is interested in composing a package with four different components: a 'Command', a 'Switch', an 'Alarm' and a 'Camera'. In ontology A these kinds of components are known, respectively, as 'Control', 'Cutout', 'Siren' and 'Photographic_Equipment'.

Suppliers Supply4 and Supply5 need to use the ontology-mapping service if they are to enter the negotiation for each of the requested components. Supply4 is the only agent who has a 'Camera' ('Photographic_Equipment' in his ontology); therefore, it is absolutely necessary that the mapping is correctly done; otherwise, Request1 will not be able to negotiate this component, which will prevent him from composing the intended package.

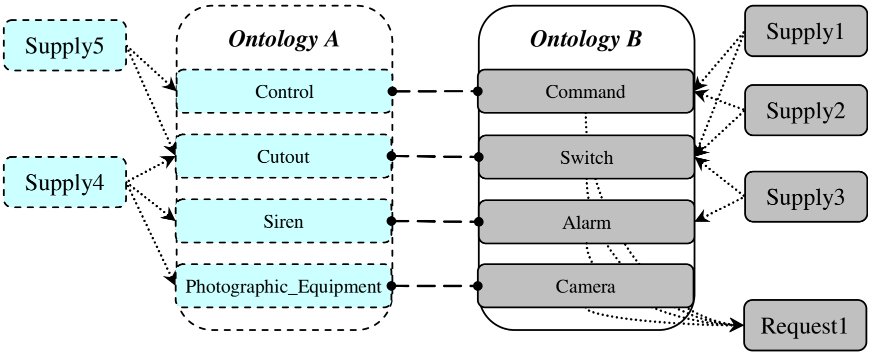


Fig. 6. Scenario: agents, ontologies and classes of components

In addition to the class names, the ontologies also differ in the attribute names for each class. Table 1 summarizes the attribute names for both ontologies. We can notice that the attribute named 'price' is the only one which is actually the same in both ontologies. For all other attribute names, some are lexically similar, while others have only a semantic resemblance. For instance, 'has_wireless' in ontology B is lexically similar to 'wireless' in ontology A. On the other hand, the attribute 'sight_grade' in ontology A has no lexical similarity with the attribute 'vision_angle' in ontology B, and yet they mean the same thing. Hence it is easy to anticipate that the mapping of the attributes 'wireless' and 'has_wireless' will be solved by using *N-Grams*, while *WordNet::Similarity* will help on solving the pairing of attributes 'sight_grade' and 'vision_angle'.

Table 1. Class attributes for ontologies A and B

Ontology A		***Ontology B***	
Attribute	*Class*	*Attribute*	*Class*
Price	*all*	price	*all*
Range	Control	reach	Command
Cipher	Control	code	Command
num_button	Cutout	number_button	Switch
Decibel	Siren	db	Alarm
Wireless	Photographic_Equipment	has_wireless	Camera
sight_grade	Photographic_Equipment	vision_angle	Camera
lens_dimension	Photographic_Equipment	lens_size	Camera

According to this scenario, we expect that the ontology-mapping service is able to map classes from ontology A with classes from ontology B. For instance, when Supply4 receives a CFP from the negotiation mediator (see Figure 4) asking for a 'Camera', he looks at his ontology and does not find that class. Consequently, he will ask the ontology-mapping service, which will give him the respective mapping, telling him that 'Camera' represents a concept similar to 'Photographic_Equipment'. Additionally, the service will also give him a mapping between the attributes of 'Camera' and those of 'Photographic_Equipment'.

4.2 Results

Table 2 shows the values obtained with the mapping process applied to the classes 'Camera' and 'Photographic_Equipment'. As we can see, the mapping between class names was obtained using *WordNet::Similarity* [17], after a foreseeable failure of *N-Grams* – there is no lexical similarity between the two words. This value is 0.81 and represents 50% of the final score for this class[1]. Attribute 'price' has a perfectly matching attribute according to *N-Grams*, hence the confidence of 1.00. Attribute 'has_wireless' had a satisfactory matching with attribute 'wireless' using *N-Grams* (0.64). As for attributes 'lens_size' and 'vision_angle', they both did not get a satisfactory result using *N-Grams*. A better result was obtained using *WordNet::Similarity*: 'lens_size' matched 'lens_dimension' with 0.85; 'vision_angle' matched 'sight_grade' with 0.73. These results were obtained by averaging the mappings of two pairs of words composing the attribute names. The global score for attribute matching is the average of each individual attribute matching score: (1.00 + 0.64 + 0.85 + 0.73) / 4 = 0.81. The final score is then the average of both (class and attributes) scores: (0.81 + 0.81) / 2 = 0.81.

Table 2. Mapping results for classes 'Photographic_Equipment' and 'Camera'

Ontology A	***Ontology B***	*Confidence*		
Photographic_Equipment	Camera	0.81 (*WN::Sim*).	0.81	0.81
price	price	1.00 (*N-Grams*)	0.81	
wireless	has_wireless	0.64 (*N-Grams*)		
lens_dimension	lens_size	0.85 (*WN::Sim*)		
sight_grade	vision_angle	0.73 (*WN::Sim*)		

The ontology-mapping service gave results of high confidence for all the classes considered. Results are summarized in Table 3.

The user interface at Figure 7 shows, for Supply4 (who had components belonging to classes 'Cutout', 'Siren' and 'Photographic_Equipment'), that all these classes were correctly mapped by the ontology-mapping service. Supply5 also had a list of mapped classes for 'Control' and 'Cutout'.

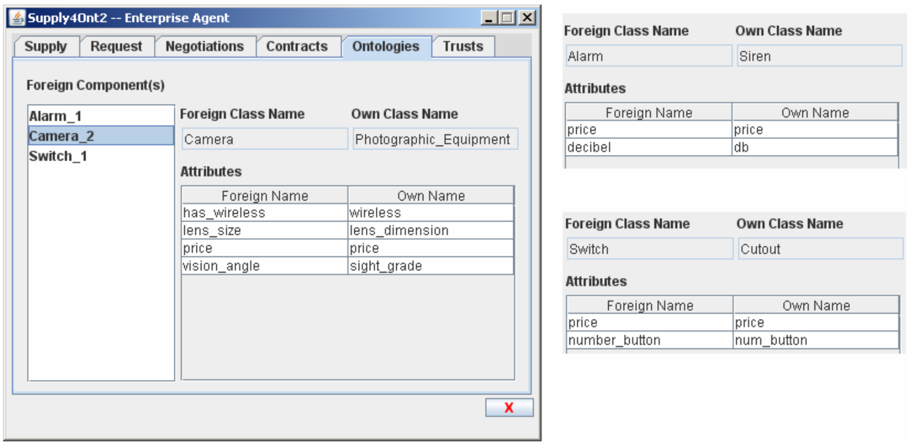
---

[1] The actual score obtained by *WordNet::Similarity* for mapping the class names 'Camera' and 'Photographic_Equipment' is 2.99, which was then normalized to the range [0.0; 1.0].

Table 3. Confidence values for each class mapping

Ontology A	***Ontology B***	*Confidence*
Control	Command	0.97
Cutout	Switch	0.82
Siren	Alarm	0.90
Photographic_Equipment	Camera	0.81

Figure 8 shows information regarding the negotiation of the four components that were part of the package intended by Request1. In the second column we can see the number of agents who negotiated those components. Looking again at Figure 6, we can conclude, that all agents who had a component to supply were involved in the negotiation, regardless of the ontology they had adopted. Three agents negotiated the 'Command' component: Supply1, Supply2 and Supply5. All agents negotiated the 'Switch' component. Two agents negotiated the 'Alarm' component: Supply3 and Supply4. Finally, the only supplier who had a 'Camera' – Supply4 (a 'Photographic_Equipment' in his ontology) – was the only one who entered the negotiation of that component.



Foreign Component(s)

Alarm_1
Camera_2
Switch_1

Foreign Class Name	Own Class Name
Camera	Photographic_Equipment

Attributes

Foreign Name	Own Name
has_wireless	wireless
lens_size	lens_dimension
price	price
vision_angle	sight_grade

Foreign Class Name	Own Class Name
Alarm	Siren

Attributes

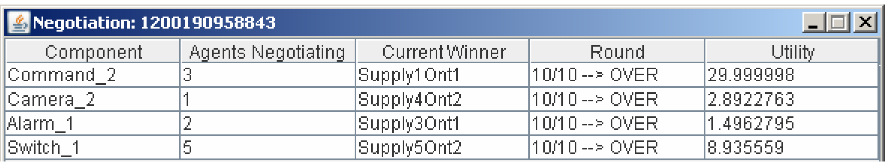
Foreign Name	Own Name
price	price
decibel	db

Foreign Class Name	Own Class Name
Switch	Cutout

Attributes

Foreign Name	Own Name
price	price
number_button	num_button

Fig. 7. Mapped classes for Supply4



Component	Agents Negotiating	Current Winner	Round	Utility
Command_2	3	Supply1Ont1	10/10 --> OVER	29.999998
Camera_2	1	Supply4Ont2	10/10 --> OVER	2.8922763
Alarm_1	2	Supply3Ont1	10/10 --> OVER	1.4962795
Switch_1	5	Supply5Ont2	10/10 --> OVER	8.935559

Fig. 8. Negotiations

The use of the ontology-mapping service made it possible for agent Request1 to successfully negotiate all the components of the package it intended to assemble.

5 Conclusions

The heterogeneity problem in ontology specification is a strong impediment to the development of interoperable automated tools. In our case, we address this interoperability issue from a multi-agent system perspective: agents need to solve their ontological differences in order to be able to automatically negotiate on behalf of their owners. Although ontology schema mapping is, in general, not likely to be fully automated, we have implemented an approach with such an aspiration in mind. However, as noted in [20], some schema semantics is often not explicit and thus cannot be automatically processed, and therefore matching tools should be used to determine match candidates, to be confirmed by the user. Also, the user should be empowered to manually specify matches that the system was not able to find, and therefore appropriate user interfaces should be developed with this concern.

The research literature devoted to Electronic Institutions does not emphasize the importance of having ontology mapping services. In this respect, our approach is original, as far as we know. Some authors [21] point out the need for having a common ontology available for all parties inside the institution, describing both general and domain-dependent concepts. These approaches therefore avoid the heterogeneity problem.

A central feature in our developments is the contextualization and integration of the ontology-mapping service with a negotiation protocol for agent-based automated negotiation. The service enables the use of such automation in open settings, which would otherwise be unfeasible. The approach to ontology mapping that we have integrated does not require an enterprise agent to reveal possibly sensitive information regarding his competencies. All is required is that he is able to describe his ontology in terms of classes and attributes. Moreover, the original ontologies can be maintained, which is an important advantage in a B2B context, where an ontology switch can be an expensive task.

Other authors have tackled the problem of ontology disparity in the past. However, most of them do not integrate their approaches with agent interaction protocols. Furthermore, some approaches force modifications in the original ontologies [22], require the inspection of instances described in those ontologies [23], impose the creation and usage of a new merged ontology [24], or assume more requirements on the original ontologies' representations [25]. An approach to ontology alignment in MAS communication is presented in [26]; in this case, an upper ontology describing general concepts that are the same across all domains is assumed to exist.

We are aware that our experiments are based on simplified artificial scenarios. In fact, most experiments reported in the literature so far, are toy problems. Real experiments with ontology mapping and integration are missing, probably caused by the lack of available real-world ontologies on the Web. The basic principle that we rely on – the fact that two different ontologies representing the same domain will describe concepts with (probably) a similar syntax and share similar attributes – leads us to believe that our approach is sensible. However, as pointed out above, this process is far from being a solution to enable automatic negotiation when using real-world ontology schemas. More likely, an improved version of this ontology-mapping tool could be used as an assistant to find potential matches.

Acknowledgments. The authors would like to thank Andreia Malucelli for her PhD work on the ontology-mapping service. The first author is supported by FCT (Fundação para a Ciência e a Tecnologia) under grant SFRH/BD/29773/2006.

References

1. Luck, M., et al.: Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). In: AgentLink (2005)
2. Davulcu, H., et al.: Modeling and Analysis of Interactions in Virtual Enterprises. In: Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises. IEEE Computer Society, Los Alamitos (1999)
3. Lopes Cardoso, H., Oliveira, E.: Electronic Institutions for B2B: Dynamic Normative Environments. Artificial Intelligence and Law 16(1), 107–128 (2008)
4. Camarinha-Matos, L.M., Afsarmanesh, H.: Elements of a base VE infrastructure. Journal of Computers in Industry 51(2), 139–163 (2003)
5. Hviid, M.: Long-Term Contracts and Relational Contracts. In: Bouckaert, B., Geest, G.D. (eds.) Encyclopedia of Law and Economics, Volume III: The Regulation of Contracts, pp. 46–72. Edward Elgar, Cheltenham (2000)
6. Castelfranchi, C.: Engineering Social Order. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2000. LNCS, vol. 1972, pp. 1–18. Springer, Heidelberg (2000)
7. FIPA-ACL. Message Structure Specification, `http://www.fipa.org/specs/`fipa00061
8. Malucelli, A.: Ontology-based Services for Agents Interoperability. Faculdade de Engenharia, Universidade do Porto, Porto (2006)
9. Willmott, S., Constantinescu, I., Calisti, M.: Multilingual Agents: Ontologies, Languages and Abstractions. In: Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents, Montreal, Canada (2001)
10. FIPA. Foundation for Intelligent Physical Agents, `http://www.fipa.org`
11. FIPA-OSS. Ontology Service Specification, `http://www.fipa.org/specs/`fipa00086
12. Lopes Cardoso, H., et al.: Institutional Services for Dynamic Virtual Organizations. In: Camarinha-Matos, L.M., Afsarmanesh, H., Ortiz, A. (eds.) Collaborative Networks and Their Breeding Environments – 6th IFIP Working Conference on Virtual Enterprises (PRO-VE 2005), Valencia, Spain, pp. 521–528 (2005)
13. Damashek, M.: Gauging Similarity via N-Grams: Language-independent Sorting, Categorization, and Retrieval of Text. Science 267, 843–848 (1995)
14. Princeton University. WordNet: a lexical database for the English language, `http://wordnet.princeton.`edu
15. Lenat, D., Miller, G., Yokoi, T.: CYC, WordNet, and EDR: Critiques and Responses. Communications of the ACM 38(11), 45–48 (1995)
16. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:Similarity - Measuring the Relatedness of Concepts. In: Nineteenth National Conference on Artificial Intelligence (AAAI 2004), San Jose, CA (2004)
17. Pedersen, T.: WordNet:Similarity, `http://wn-similarity.sourceforge.net/`

18. JADE. Java Agent DEvelopment Framework, `http://jade.tilab.com`
19. Stanford Center for Biomedical Informatics Research. The Protégé Ontology Editor and Knowledge Acquisition System, `http://protege.stanford.edu`
20. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The International Journal on Very Large Data Bases 10(4), 334–350 (2001)
21. Dignum, F.: Agents, markets, institutions and protocols. In: Dignum, F., Sierra, C. (eds.) Agent Mediated Electronic Commerce: The European Agentlink perspective, pp. 98–114. Springer, Heidelberg (2001)
22. Bailin, S.C., Truszkowski, W.: Ontology Negotiation between Scientific Archives. In: Thirteenth International Conference on Scientific Statistical Database Management. IEEE Computer Society Press, Fairfax (2001)
23. Wiesman, F., Roos, N.: Domain independent learning of ontology mappings. In: Third International Joint Conference on Autonomous Agents & Multi Agent Systems. ACM Press, New York (2004)
24. Pinto, H.S., Gómez-Pérez, A., Martins, J.P.: Some Issues on Ontology Integration. In: Workshop on Ontology and Problem-Solving Methods: Lessons Learned and Future Trends. CEUR Publications, Stockholm (1999)
25. van Eijk, R.M., et al.: On Dynamically Generated Ontology Translators in Agent Communication. International Journal of Intelligent Systems 16, 587–607 (2001)
26. Mascardi, V., Rosso, P., Cordì, V.: Enhancing Communication inside Multi-Agent Systems - An Approach based on Alignment via Upper Ontologies. In: Baldoni, M., Baroglio, C., Mascardi, V. (eds.) Int. Workshop on Agent, Web Services, and Ontologies Integrated Methodologies (MALLOW-AWESOME 2007), Durham, pp. 92–107 (2007)