8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
Suceava, Romania, May 25 – 27, 2006

DYNAMIC ONTOLOGY MAPPING FOR E-COMMERCE

Ioan Alfred LETIA[1], Mihai COSTIN[2]
Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-400391 Cluj-Napoca, Romania
[1]letia@cs.utcluj.ro

Abstract. *In order to have a functionally and usable e-commerce system, the first obstacle that has to be surpassed is dealing with the different views of all partaking actors or agents and this can be done by means of mediation[1]. This mediation, which has as purpose solving different mismatch problems, must take place at the ontology level of the different sides in order for them to be able to share information. This paper will emphasize the idea that in an e-commerce environment the mediation must take a dynamic form in order to be successful.*
Keywords: *ontologies, mismatch, MAS, dynamic mapping.*

1. Introduction

For information systems, homogeneity is hard, if not impossible to achieve. The main reason is that all existing systems use, explicitly or implicitly, their own ontology and it is quite unlikely that the wide variety of these ontologies will conform to a single standard because of multiple reasons, both technical and economical. Still, it would be desirable, especially from the semantic web approach, for heterogeneous components to be able to collaborate and to exchange information regardless of the underlying tiers. A large array of applications, like the ones in the e-commerce or e-government category, would benefit mostly from this transparent collaboration and information sharing. Even more, as pointed out in [5], this ability is essential for MAS systems, which have the strongpoint of partitioning the problem space and assigning a piece to different agents with different knowledge and representations.

A great amount of research is involved in this area and more and more methods are proposed to solve the problem or aspects of the problem, including theoretical methods that unify different approaches in order to find the best solution [3].

As mentioned, the differences between two or more such heterogeneous components (for example, in a MAS e-commerce application, the agents that represent the different part-taking actors) are situated at ontology level. These differences are called ontological mismatches.

The first step when dealing with an ontological mismatch is to define that type of mismatch, followed by choosing the appropriate method of solving the problem.

This mismatch problem for the ontologies can be an obstacle at least in two common scenarios in the semantic web context: ontology reuse and ontology based-communication. Ontology reuse is one of the main points when dealing with developing ontologies and can be considered as important as the reusability for an object oriented approach. The main concern in this case is for one to be able to enrich an existing ontology by adding new concepts or to create new ontologies by means of composition or aggregation. Ontology based-communication tackles the problem of two or more entities that rely on top of one or more ontologies and are trying to exchange information. In most of the cases, the mentioned ontologies are not directly available and can be quite different (in fact, agents that have different ontologies would have more potential information to exchange than the ones that use exactly the same ontologies).

[1] The term mediation is used here in an ontology context and defines actions as mapping or merging, which solve ontology mismatch problems.

From the point of view of e-commerce, the second one presents more interest since, as pointed out before, the main problem in this area is the lack of homogeneous agents (and implicitly, ontologies that support this agents) taking part in the B2B or B2C process.

1.1 Ontology mismatch

Two or more ontologies may exhibit conflicts at ontology level (semantic level) or at language level (syntactic level).
The mismatch at language level can be caused by different representing language (for example, one ontology using OWL2 and another one using RDF3). The mismatch at semantic level can be caused by multiple reasons in a range from ontologies that have different representations for the same concepts to ontologies that don't refer to the same domain, but the main source of conflict is the use of conflicted or mismatched terms about concepts [6].
These mismatches can be solved by using methods that include aligning, mapping, translating, merging or integration. The method to be used must be chosen based on the source and target ontologies, on their availability and eventually on the domain that is being tackled.
For the case considered in this paper the general domain is taken to be e-commerce, although any other context can suite just as well. A much more important aspect is the fact that the ontologies are not considered to be public available, as they are in [1], and can be accessed only trough the agents that rely on them, these agents representing the entry points to the information held at ontology level. Practically what the proposed scenario is trying to do is to map a real world (economic) context in which information is quite precious and not always available, communication is done by exchanging pieces of information, or tokens [4] and not the whole repository, and the facade to the information repository are one or more agents that have more or less the purpose of controlling the flow of information.
In this kind of context most of the classical 'whole-ontology' merging and mapping solutions are prone to failure, and thus a form of dynamic mapping must be imposed to solve the problem.

2. Problem Statement

2.1 Scenario

The problem that this paper is trying to solve can be more easily described by the following scenario:
In an e-commerce environment (modelled as a MAS) there are different actors that are taking part to the process. In this case three actors have been chosen. There is a front agent that takes contact with the client (this client can be human or machine), and will be called client-agent and two other actors that represent two different factories that can produce goods and will be called producer-agent-one and producer-agent-two. These agents can communicate using a wide accepted communication language that will be allowing them to exchange tokens with no regard to the underlying ontology.
The client-agent has an ontology describing the services it can provide to its clients and according to the chosen scenario it has knowledge of furniture components and arrangements (like, for example living room furniture composed out of a table and four chairs).
The producers have their own ontologies (catalogs) that are assumed not to be public available, the agents being the only interface to the data stored in those ontologies and have knowledge of the goods that the factory can produce, goods named by the factory's own procedure (a real life example of such a catalog would be RosetaNet or UNSPSC4, catalogs which are widely used at the moment by different actors).
For example, we can assume that the element representing a chair it is called “chair” in the client-agent ontology and “CHx” in the

[2] http://www.w3.org/OWL
[3] http://www.w3.org/RDFS
[4] http://www.unspsc.org/

producer-agent-one's ontology. Both of these elements have attributes and relations that define it with respect to the owning ontology. Furthermore, the producer has no knowledge of or interest for different furniture arrangements present on the client, so in this case the client is the one that must query the producers for different components of an arrangement.
Since the ontologies are not public available (the agent's can't look into each other's heads to find out what the other knows) the only way of information exchange can be done through agent communication. This communication will help the agents to establish a partial mapping for the elements of interest in order for their collaboration to take place with success. The partial mapping is an important aspect of the way the agents interact since mapping the whole ontologies, besides being computationally expensive, maybe impossible in the case of ontologies referring to different domains.

We also presume that, even if the ontologies would be public available, the client can't have in an efficient way a complete mapping from its elements to the elements of the producer from another source out of a couple of reasons: 1) the amount of data that would have to be carried after the client 2) Adding a new producer and removing an old one from the network would force the client (clients) to rewrite all his (theirs) mappings to be up to date.

2.2 Goal

The described problem leads to the idea that some form of dynamic mapping has to take place between the agents (their ontologies) if communication is to take place with success and that the agent will be updating its own knowledge as more and more queries are answered.
Also, some form of discovering the other agents, language to be used and methods of integrating new knowledge have to be found along with the mentioned mapping between the agents.

3. Information exchange and integration

The proposed solution to the problem we are facing involves managing information sharing and exchange in a multi agent system in which each agent uses an ontology to represent its knowledge.
These agents can receive queries (from another agent) that must be solved and can engage (be engaged) in a 'discussion' with another agent in order to acquire knowledge if the present available knowledge can't solve the query.
The problem appears when an agent is trying to acquire more knowledge and that knowledge is held by another agent that uses a different ontology behind causing a conflict to appear when the two are trying to communicate. The communication takes place with the help of information tokens exchange between the agents, information tokens that will also represent the main mean of obtaining the partial mapping needed for the actual information sharing to take place. The idea of tokens is similar to the one in [4], but the approach proposed here will not be using these tokens in order to create an explicit information channel or a global ontology that would serve the two semantically integrated agents. Instead, those tokens will be used, somehow similar to real life exchange of information between humans, in order to achieve a lazy dynamic mapping. The mapping is both lazy and dynamic because the communication will have as purpose the translation of only a couple of concepts from the ontologies and only when those concepts are needed.
Another possible approach to the stated problem could be similar to the one described in [2], that is also tackling the problem of agents based on heterogeneous ontologies, but the present paper is aiming at also enriching the agent's knowledge beside making the communication possible between the entities and is taking a more common-sense (real life based) approach to the problem of asking and answering queries.
The idea behind the proposed mapping solution is based on the fact that, for a given domain, ontologies are more or less created by humans and all elements or some of their attributes have attached a textual description. (This idea is similar to the grounding ontology theory [5] but in this case, the common underlying structure is more transparent and is given exactly by the

assumption that at the base of an ontology specification there is a human factor). This description along with the attributes of an entity and the relations of that entity with other entities represents the information contained by the ontology. When two agents are trying to exchange knowledge they will exchange tokens of the mentioned information. These tokens will be used to answer the queries or to integrate knowledge into the existing ontology.

When receiving a token, the agent must take into account the context of the token in order to decide what action to take - for example if a query was made using that token or an answer has been given that contains the token. In each case, the token may be the only element that is unknown to the agent, and if that happens, a translation must be made in order for the interaction to proceed. This translation or mismatch solving will be tackled in the next section.

For the agents to be able to communicate a set of common vocabulary is required. This vocabulary must be accurate enough while using the smallest set of concepts [5]. A wide accepted agent communication language (or a subset o it) can be used (like FIPA ACL or KQML), although this languages don't specify the semantics of the content but only the syntax. Alternatively a common communication language can be constructed and used by the agents like in [5] and for the proposed scenario we will use a simple common communication language with elements (predicates) such as ask, buy, confirm...

We could imagine the following flow:

The client-agent must get a set of furniture for one of its customers composed out of 4 chairs and a table. First, the agent will check its own ontology to see if he knows who is producing such components. If it founds something will ask for a confirmation from those agents and will present the offer to its client. If one or more components don't have a producer in the knowledge base or a confirmation is not made, the agent must find a producer for that component, or if we are to make a parallel with the OO world, the agent must find an implementation for one of its abstract classes.

Let's say that the chair element has no producer in the client-agent knowledge base. Now the agent must find all the producers (beside the ones it already knows) and ask them about the chair token. For finding the producers we can imagine, for example, a network using JINI5 technology where all the agents that have joined that network can be discovered after the services they provide. After all the producers have been found the agent will send a query to them, asking about the chair product. If the produces have no knowledge of the chair element will ask for a clarification and so the translation (mismatch solving) process will begin. After the mismatch has been cleared the producer will send its term for the chair along with all the attributes to the client-agent which will now have more knowledge about the chair component and will be able to present an offer to its customers.

4. Mismatch solving

As mentioned in the previous section, a mismatch has appeared between the two agents regarding the "chair" element.

The two will now try to solve that mismatch using the following predicate exchange:

```
Client-agent -> producer-agent:
      query(produce, chair)
Producer-agent -> client-agent:
      clarify(chair)
Client-agent -> producer-agent:
      startExplainToken(chair)
      token("Furniture element with 3
or 4 legs")
      token("made out of wood or
metal")
      token("used by people to sit on")
      endExplainToken(chair)
```

The predicates used are known by both agents and belong to the common language used by the agent network to communicate. At each step a message check will occur, making sure that the received predicate is syntax-valid.

When the *startExplainToken* message is received the agent will begin listening for incoming tokens that have the purpose of clarifying the source of mismatch.

[5] http://www.jini.org/

The tokens sent by an agent in order to explain a mismatched concept will most likely include also tokens describing super-concepts in a similar way to the method used in [6] for retrieving a concept's meaning from the WWW. (The process can also be done in an iterative way, for large to very large taxonomies, by passing to the other agent the tokens from one conceptual level each step until the match is found)
These tokens will then be used by the receiving agent in a classification process in order to find the category/product that the initial token was referring to.
The taxonomies that represent the base structure for the ontologies belonging to the two agents could be like the ones in the following images:

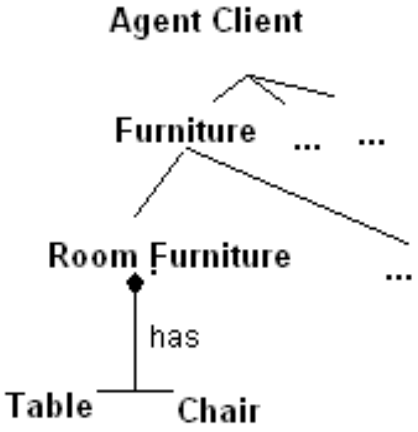

Figure 1. The client agent taxonomy.

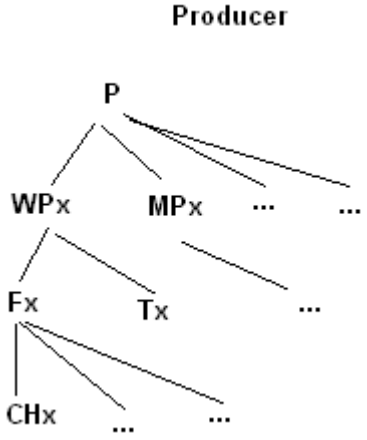

Figure 2. The producer agent taxonomy.

These taxonomies will be used by the two agents as the foundation for their ontologies, and so, for example the client agent (figure 1) will have as part of its ontology the rule

```
RoomFurniture=has(Table,      1)      ∩
has(Chair, 4 )
```

Also, both of the agents will have, as mentioned before, beside other attributes, textual descriptions for the elements or the categories belonging to the ontologies.
In the considered scenario we will take into account the taxonomies (as the taxonomies stand at the base of any ontology definition) and the textual descriptions for the two agents in order to exemplify the proposed process of mapping.
The producer agent will have attached in its ontology attributes for each element and so, for example, it will have a description element for *WPx*, d="Products made out of wood", for *Fx*, d="Furniture elements", and for *CHx*="Chairs with 4 legs" and so the producer-agent will be able to find the match of the questioned token in his ontology by using the provided information by the client-agent. After the element was found it will be sent, along with some attributes, to the client-agent and now this agent will have an extra piece of knowledge regarding the chairs produced by the producer-agent allowing it to make the future interactions much more easy (lazy dynamic mapping).
As mentioned, the received tokens will be used by a text classificator to find the category and then product that meets the given description.
These received tokens can be viewed as a document that must be classified in one of the existing categories at destination. There are means of including in this document (and in the classification process) also other attributes beside description, like key-value pairs, but here we treat just the simple case in which the exchanged tokens contain only textual descriptions. For this classification processes a naive bayesian classifier together with a synonyms dictionary (like for example WordNet[6]) will be used as described below.

[6] http://wordnet.princeton.edu/

Regarding the usage of the word dictionary, more complex relations between words and concepts can be used beside synonymy, like hyponymy and hypernymy that can define parent – child ("is a") relationship between concepts or holonymy and meronymy ("has a" relations) in order to have a more flexible concept matching and classification.
The naive bayesian classifier will estimate the posterior probability of the token-based document belonging to category C_i using that document as evidence:

$$P(C_i|d) = \frac{P(Ci) * P(d|Ci)}{P(d)} \quad (1)$$

In the given equation P(d) can be ignored since it is the same for all the destination categories (D) and $P(C_i)$ can be estimated as :

$$P(C_i) = \frac{\text{Nr. of elements in Ci}}{\text{Total nr. of elements for D}} \quad (2)$$

The only remaining term to be computed is $P(d|C_i)$. If we take a simplistic approach by assuming that the words are independent from each other, this probability can be estimated using the following equation:

$$P(d|C_i) = \prod_{w \text{ in } d} P(w|Ci) \quad (3)$$

where **w** represents the words that are part of the document **d**. In fact, **w** represents more that just one word, being a group of words that are all synonyms. In order to estimate $P(w | C_i)$ we can count the number of occurrences of word **w** (or any of it's synonyms) in all the descriptions of the elements from category C_i ($nrw(w, C_i)$) and divide it by the number of total words in those descriptions ($nrw(C_i)$). In order to avoid the problem caused by a word that doesn't appear in any of the descriptions of the elements from category C_i (causing the number of occurrences to be zero, $nrw(w, C_i)=0$) we can use Lidstone's law of successions and so, the searched probability is:

$$P(w|C_i) = \frac{nrw(w, Ci) + \lambda}{nrw(Ci) + \lambda * |V|} \quad (4)$$

where |V| represents the size of the used vocabulary (the number of all the words from all the descriptions in all the categories), and $\lambda > 0$, whose optimal value must be chosen in order for the model to be as accurate as possible (usually this values is found by running a couple of trials on some data sets).
At this step we can compute the value of P(Ci|d) and we can choose the category with the highest probability as being the one that the element belongs to.
One issue still remains, that of choosing the categories Ci, and one approach is the following:
We do the classification process in iterations going top-down in a "divide et impera" way. At each step we choose a couple of top categories that will have all the children and parents descriptions passed onto them and run the classification process. The category/element that has the highest probability and the ones that are very close (determined by an error factor ε) to that probability will then be chosen to represent the base for the next step. At the end of this process the category/element(s) with the highest probability will be chosen as the result.

5. Conclusions and future work

In this paper we have pointed out the importance of finding a solution for the ontology mismatch problem in a MAS environment and that a "classic" whole-ontology mapping approach is not feasible in this case. We have proposed, as an alternative to the whole-ontology mapping, the lazy dynamic mapping in order to achieve the desired communication level between the agents in such a network. This mapping is based on token exchange, in a similar way to the one presented in the described scenario, between two agents followed by a matching process at the receiver's side, a matching process that is using a naïve bayesian classificator.
The solution presented here is only the first step in solving the problem as the approach taken here is a simplistic one, and can be viewed as the starting point in constructing a real framework that will be able to solve the

mismatch problems for ontologies, as communication takes place between agents, by using a form of dynamic mapping appropriate to the MAS environment.

Some of the possible future enhancements to the proposed process would be to add a formal communication vocabulary as defined in [5] in order to enhance the communication abilities of the partaking agents, and, directly related to the presented mapping method, to introduce bayesian networks in order to improve the mapping process [6].

Another important issue that is to be taken into account, especially for the implementation phase, is the one related to performance and scalability when mapping and classifying in large taxonomies or with many actors that take part to the process.

Also, a more elaborate scenario and study case will be used in order to capture more of the aspects present in a real world MAS.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. (2004) *On integrating catalogs*.

[2] Patrick Doherty and Witold Lukaszewicz. (2004) *Approximative query techniques for agents with heterogeneous ontologies and perceptive capabilities*. In The 9th International Conference on Principles of Knowledge Representation and Reasoning.

[3] York Sure, Marc Ehrig. (2004) *Ontology mapping - an integrated approach.*

[4] Marco Schorlemmer and Yahhis Kalfoglou. (2004) *Progressive ontology alignment for meaning coordination: An information-theoretic foundation.*

[5] Jurrian van Diggelen, Robert Jan Beun, Frank Dignum, Rogier M. van Eijk and John-Jules Meyer (2005). *Optimal Communications Vocabularies and Heterogeneous Ontologies*

[6] Zhongli Ding, Yun Peng, Rong Pan, Yang Yu (2005). *A Bayesian Methodology towards Automatic Ontology Mapping*