Ontology Mapping in Open Multi-Agent Systems

Mihaela Oprea

Department of Informatics, University Petroleum-Gas of Ploiesti

Bd. Bucuresti Nr 39, Ploiesti, 100680, Romania

mihaela@upg-ploiesti.ro

Abstract:. In a multi-agent system the agents communicate between them in order to fullfil the global goal or their local goals. When the system is an open one, i.e. the agents enter and leave the system dynamically, the problem of ontology heterogeneity becomes more important and has to be solved. In order to communicate, agents need to share the same ontology (totally or partially) or at least common concepts that are synonyms. Thus, an ontology mapping mechanism has to be included in the system. The paper presents some ontology mapping methods that were reported in the literature, and focus on the application of `OntoMap`, an ontology mapping mechanism that we have proposed in a previous work, in the case of an open multi-agent system such as `VIRT_CONSTRUCT`, an agent-based virtual enterprise that we have developed in the housing domain.

Keywords: ontology mapping, multi-agent system, lexical method, structural method.

Mihaela Oprea received her MSc degree in Computer Science from University Politehnica Bucharest in 1990 and her PhD degree from University of Ploiesti in 1996. Currently, she is a full professor at the Department of Informatics of the University Petroleum-Gas of Ploiesti. Her main research interests include pattern recognition algorithms, machine learning, knowledge modelling, applications of multi-agent systems and artificial intelligence techniques in various domains such as environmental protection, engineering, and education. She has published more than 70 research papers in the area of artificial intelligence in international journals, and in the proceedings of international conferences and workshops. Since 1995 she has visited as a research visitor universities and artificial intelligence research institutes from UK, Austria, Spain, Greece, Sweden, Czech Republic, Hungary, and France. Dr. Oprea is a member of the Romanian Society for Automatics and Technical Informatics (SRAIT), the Slovenian Artificial Intelligence Society, the International Society for Machine Learning (ISML), and the IASTED Technical Committee on Artificial Intelligence and Expert Systems.

1. Introduction

One of the key challenges for a real-world application of multi-agent systems technology in the Internet environment is to solve the interoperability problem [10]. More specifically, to find mechanisms by which intelligent agents that work in open environments could communicate in an efficient way, even in the cases when they use completely different ontologies. A solution would be to build common big ontologies that could be shared by agents. Such a solution is not realistic yet, and moreover, cannot obtain the agreement of the majority of specialists in their domain of expertise. Another solution would be to design specialised agents that are doing ontology mapping when such a task is asked by an intelligent agent. Yet, another more efficient solution, would be to include an ontology mapping mechanism in the architecture of each agent. In this paper, we focus on the topic of ontology mapping in the case of open multi-agent systems, in which agents can enter and/or leave dynamically, and take as an example the agent-based virtual enterprise, `VIRT_CONSTRUCT` ([15], [16]), for which we have designed an ontology mapping mechanism *`OntoMap`* (presented in detail in [14]) that is included in the architecture of each agent.

The paper is structured as follows. In section 2 we formulate the ontology mapping problem and briefly discuss the current methods that are used. Also, it is presented our previously proposed ontology mapping mechanism, *`OntoMap`*. In section 3 it is described a case study of ontology mapping for the agent-based virtual enterprise `VIRT_CONSTRUCT`. Some experimental results are presented in section 4. Section 5 concludes the paper and highlights the future work.

2. Ontology Mapping

An ontology is a specification of a conceptualization [6]. The development of an ontology involves the definition of all application domain specific terms (i.e. concepts, relations and instances). Also, a set of axioms that restrict the use of the terms has to be defined. The concepts are classes in the ontology and are organized in taxonomic hierarchies (trees), while the properties (attributes) of concepts are slots in the ontology and the restrictions are facets of the slots. A concept may be defined as an aggregation of attributes and sub-concepts.

Formally, we could define an ontology O as follows:

$O = \{\mathcal{V}, \mathcal{D}, \mathcal{A}, \mathcal{T}\}$,

where $\mathcal{V}$ is the vocabulary of terms, $\mathcal{D}$ is the set of definitions for all terms, $\mathcal{A}$ is the set of axioms, and $\mathcal{T}$ is the set of ontology trees. The non-terminal nodes of an ontology tree are concepts, while the leaf nodes could be concepts or instances. The arc between two nodes represents the relation between two concepts or between a concept and an instance or between two instances.

In a multi-agent system the ontology that is used in a communication depends on the subject of the communication. In some cases a standard ontology (such as Dublin Core) can be used, while in most of the cases it is an ontology dependent on the application domain. Moreover, the terms specific to a domain can be described by different ontologies which could be structured in different ways, so the problem of ontology mapping becomes an important one especially in open multi-agent systems, where different types of agents with heterogeneous ontologies could enter and leave dynamically.

2.1. Problem Formulation

There are several definitions given in the literature for ontology mapping, most of them being equivalent. As defined in [9] ontology mapping could be viewed as a learning process to find a morphism between the concepts of some given ontologies. According to [3] the ontology mapping of two ontologies means that for each entity (concept, relation, or instance) in one ontology, we try to find a corresponding entity, which has the same intended meaning in the other ontology. In this paper we shall use this last definition that is formulated as follows.

Let's consider two ontologies, O_1 and O_2. A mapping between these ontologies can be defined as a partial function *map* that finds the maximal number of potential mapping pairs (t_1, t_2), where t_1 and t_2 are terms from O_1 and O_2, respectively.

$map(O_1, O_2) = \{(t_1, t_2) \mid t_1 \in O_1, t_2 \in O_2\}$

So, t_1 is the term expressed in O_1 and t_2 is its translation in O_2. The function *map* is not necessarily one-to-one. If there are no overlapping concepts in O_1 and O_2 then no mapping can be found. Figure 1 shows the picture of ontology mapping between two generic ontologies. In this case $map(O_1, O_2) = \{(t_1, t_2), (u_1, u_2)\}$.

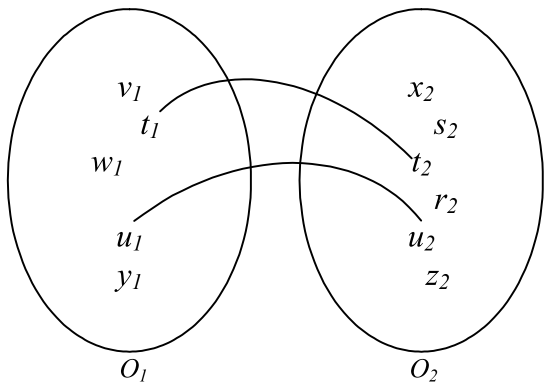

Figure 1. Ontology mapping

Figure 2 shows an example of ontology mapping between two ontologies with generic terms from the domain of virtual enterprises.

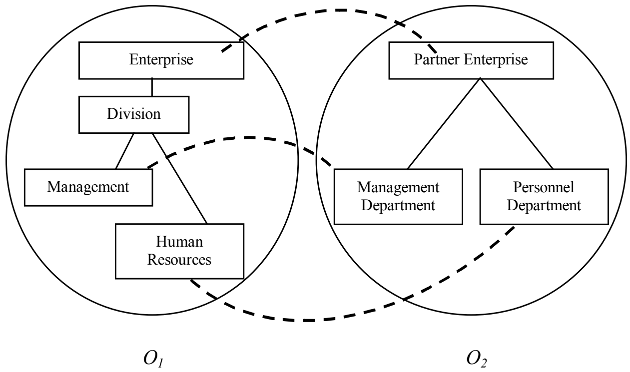

Figure 2. Example of ontology mapping in the domain of virtual enterprises

Different formulations for similar terms (i.e. words with the same meaning), could appear in the two ontologies due to different semantic structures (structural conflicts), different names for the same type of information or the same name for different types of information (type conflicts), and different representations of the same data (data conflicts). The structure of an ontology could be flat or hierarchical. When two ontologies have completely different hierarchical structures, the structural conflict becomes more severe.

2.2. Ontology Mapping Methods

In the recent years, various ontology mapping methods and tools have been reported in the literature (see e.g. the review from [7]). The main conclusion of the research done so far is that the development of fully-automated ontology mapping tools is very difficult, and thus, human validation is still needed.

Currently, there are two main classes of ontology mapping methods: (1) lexicon-based methods, and (2) structural methods.

When the natural language used in the ontologies that are mapped is the same (e.g. English) the probability that the ontologies bear lexical similarity in their vocabularies describing the same concepts is higher. Therefore, in such a case a lexicon-based method is feasible and more appropriate. In [9] it is described a semi-automatic lexicon-based ontology mapping tool, LOM, that uses four matching methods: whole term matching, word constituent matching, synset matching, and type matching. LOM provides as output a list of the matched pairs of terms with scores ranking their similarity. The main problems encountered when using a lexicon-based method are given by abbreviations, proper names, shorthands, codes and abstract symbols (e.g. in mathematics, chemistry, physics, medicine). The structural mapping methods try to solve such situations, and also the cases in which a lexicon-based mapping method is not enough. In [3] it is discussed the efficiency of ontology mapping methods focusing on some well known methods reported in the literature [7]: NOM (Naïve Ontology Mapping), PROMPT, GLUE and QOM (Quick Ontology Mapping, [4]). The best complexity is given by QOM, $O(n \cdot \log(n))$, where n is the number of leaf concepts in the ontology hierarchy. Among the structural approaches that could be applied to ontology mapping we could mention: GLUE (described in [7] - uses a machine learning technique), Anchor-PROMPT (an advance version of PROMPT [13] – uses similarity measures based on ontology structures), fuzzy syntactic analysis [7]. In [11] it is presented an efficient clustering mechanism, while in [2] are shown the main benefits of ontology mapping when using conceptual graphs for the representation of personal ontologies. A heuristic mapping method for semantic enrichment is presented in [18]. The two types of ontology mapping methods, lexicon-based and structural could be combined in hybrid methods. We have proposed in [14] such a hybrid method, `OntoMap`, that will be briefly described in the next section.

As most of the methods need to check the meaning of terms, we shall briefly discuss about the synset matching that is done by using WordNet [12]. WordNet is the most large, general purpose, machine readable and public available thesaurus developed at Princeton University [20], which includes about 10^6 concepts. Words in WordNet are grouped on the basis of their part of speech and organized in taxonomies where each node is a set of synonyms (called synset) representing a single sense. Some terms are not included in WordNet (e.g. proper names). When doing ontology mapping the pair of terms that have the largest number of common synsets are recorded and presented to the user for validation. WordNet is an important tool that could be used when doing synset matching, i.e. exploring the semantic meaning of the words by searching for synonyms. Some ontology mapping methods that are using WordNet are described in [3], [7], and [9].

2.3. OntoMap

Our proposed ontology mapping method combines string matching with synset matching and a very simple structural mapping method that take into account the direct concept neighbours and their relations with the analysed term. The sketch of the algorithm (applied when a discrepancy is encountered, i.e. a different term) is the following:

1. **do** string_matching()
2. **do** synset_matching()
3. **do** structural_mapping()

The first two steps of the ontology mapping mechanism correspond to a lexical analysis, and a semantic analysis, respectively. If an unknown term is encountered by an agent then he will extract the word constituents (by eliminating the connecting words or signs like *by*, *and*, *of*, _) and after that it will do string matching followed by a synset matching based on a dictionary of synonyms. When doing synset matching the ontology mapping mechanism will use the relations with the direct neighbour concepts. The similarity between terms when doing string matching is computed by using the Levenshtein distance [8]. During the synset matching it is used WordNet, and as a semantic similarity the distance between the two words in WordNet (measured in nodes). The structural similarity is computed by the distance between the neighbourhood description vectors of the two terms. If after the three steps the solution is not found, human validation is asked in order to select the correspondent term from the list of the most similar terms with the given term. The *OntoMap* algorithm is described in detail in [14].

3. A Case Study

Let's consider the case of an agent-based virtual enterprise. As stated in [1], a virtual enterprise (VE) is a temporary consortium of enterprises that strategically join skills and resources, supported by computer networks to better respond to a business opportunity. As the VE is composed by autonomous, heterogeneous and distributed entities, it could be modelled in a natural way as a multi-agent system (see for example [1] and [15]). We call a VE that is modelled as a multi-agent system, an agent-based virtual enterprise. In such a VE, each partner has its own specific ontology, and the whole VE must have an ontology which covers the specific ontologies of the partners (at least in terms of similar concepts, i.e. synonyms). As a testbed we have used `VIRT_CONSTRUCT` [15], an agent-based VE from the housing domain. In the architecture of `VIRT_CONSTRUCT`, each partner has an ontology composed by two subontologies (as shown in Figure 3), one with generic terms specific to an enterprise as an organization, and one with terms specific to the working domain, i.e. the domain of expertise.

Generic Ontology		***Specific Ontology***	
Partner	Supplier	Construction	Bricks
Department	Customer	House	Tube
Task	Working Plan	Designer	Pipe
Resource	Product	Carpenter	Machine tool
Equipment	Bank	Wood	Hammer
Worker	Account	Cement	Shovel
Order	Payment	Sand	Pump
Due time …	Location …	Plaster …	Paint …

Figure 3. Examples of terms from the ontology of a partner enterprise

In order to make easier the ontology mapping process we have developed in Protégé-2000 [17] a generic ontology for a VE, the `ABVE_Ontology` starting from previous work reported in the literature by different research groups (see e.g. [5], [19]). This ontology is updated continuously with synonyms and new terms. The basic terms defined in the `ABVE_Ontology` are actually the terms from the generic ontology of each partner enterprise. Figure 4 shows a screenshot with a sequence of the `ABVE_Ontology` hierarchy as developed in Protégé.

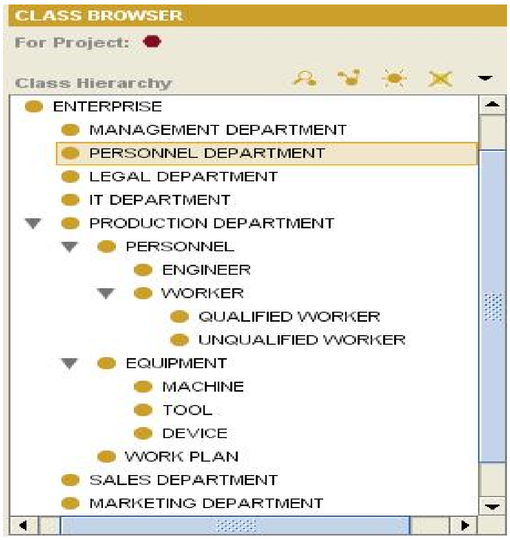

Figure 4. Sequence from the class hierarchy of the `ABVE_Ontology` (defined in Protégé)

The main types of the basic relations that appear in the `ABVE_Ontology` are *is_a* (isa), *a_kind_of* (ako), *has*, *a_part_of*. Other relations are specific to the VE domain of application.

The two ontologies used in this case study are as follows: O_1 used by `VIRT_CONSTRUCT`, and O_2 used by an agent that represent a potential partner of the VE. Figure 5 shows a part of the ontology tree of O_1, and figure 6 shows a part of the ontology tree of O_2.

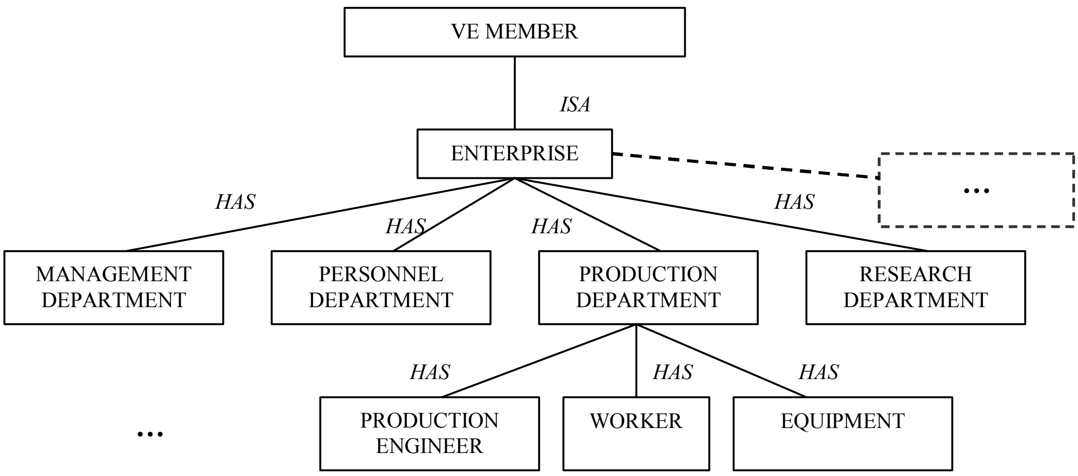

Figure 5. Sequence from the ontology tree of O_1

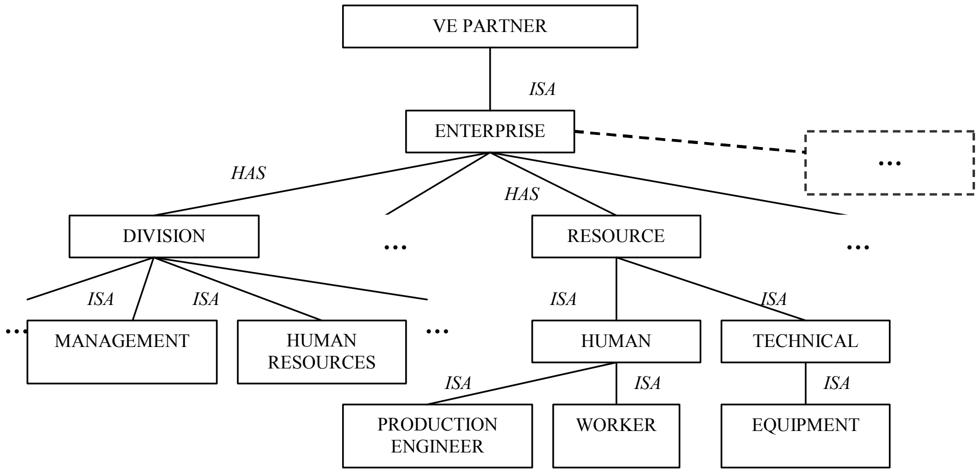

Figure 6. Sequence from the ontology tree of O_2

In our case, the two ontologies are hierarchical and have different structures. The OntoMap mechanism, described briefly in section 2.3, will find for example the following similar terms:

(VE PARTNER, VE MEMBER) – synset matching based on the dictionary of synonyms

(HUMAN RESOURCES, PERSONNEL DEPARTMENT) – synset matching based on the dictionary of synonyms for DIVISION and DEPARTMENT, and the taxonomic relation ISA from O2.

4. Experimental Results

The ontology mapping mechanism, `OntoMap`, that was included in the agent-based VE `VIRT_CONSTRUCT`, was experimented for different ontologies used by the partners of the VE, either generic or specific to the task of building construction, and/or with different hierarchical structures. The quality of the ontology mapping mechanism was measured by using two metrics: the precision and the recall given bellow.

$precision = N_1 / N_2$

$recall = N_1 / N_3$

where N_1 is the number of found mappings that are correct, N_2 is the number of found mappings, N_3 is the number of existent mappings.

The precision metric determines the fraction of the automatic discovered mappings that are correct, while the recall metric determines the fraction of the correct matches that have been discovered during the mapping process.

Table 1 summarizes the results obtained so far. We have used five ontologies with different hierarchical structures: O_1 and O_2 generic ontologies that includes terms specific to the organization of an enterprise, O_3, O_4, O_5 ontologies with generic and housing domain specific terms. The results are good, but they could be improved for example, by extending the structural analysis to more neighbours of a term, not only to the direct linked ones. We have to specify that the terms that were included in the experimented ontologies are specific to enterprises, and moreover to those that work in the housing domain, they were specified in English, and have not used abbreviations. Also, the use of the dictionary with synonyms helped the ontology mapping process.

Ontologies	*Precision*	*Recall*
O_1, O_2	0.97	0.91
O_3, O_4	0.84	0.82
O_3, O_5	0.91	0.87

Table 1. Experimental results

5. Conclusion and Future Work

In open multi-agent systems ontology mapping is one of the important issues that has to be solved in order to allow interoperability. Depending on the application domain and the complexity of the ontologies two classes of ontology mappings methods could be applied: lexicon-based and/or structural. We have developed a hybrid ontology mapping mechanism, *OntoMap*, that combines a lexical mapping with a structural mapping. Our mechanism was applied to ontology mapping in the case of an open multi-agent system such as an agent-based VE, VIRT_CONSTRUCT, that was developed by us in the housing domain. The experimental results showed a good behavior of the ontology mapping mechanism that combines a lexicon-based method with a simple structural analysis of the direct neighbours of the concepts and their relations. As a future work, we will extend the structural analysis to more neighbours, not only the direct ones.

REFERENCES

4. CAMARINHO-MATOS, L.M., AFSARMANESH, H., **Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-agent Systems Approaches**, Multi-Agent Systems and Applications, M. Luck et al. (Eds): ACAI 2001, LNAI 2086, Springer, 2001, pp. 335-364.

5. DIENG, R., HUG, S., **Comparison of Personal Ontologies Represented through Conceptual Graphs**, Proceedings of ECAI'98, 1998, pp. 341-345.

6. EHRIG, M., STAAB, S., **Efficiency of Ontology Mapping Approaches**, University of Karlsruhe, research report, 2004.

7. EHRIG, M., STAAB, S., **Quick Ontology Mapping with QOM**, University of Karlsruhe, Institute AIFB, technical report, 2004.

8. FISCHER, K., MÜLLER, J.P., HEIMIG, I., SCHEER, A., **Intelligent Agents in Virtual Enterprises**, Proceedings of the 1st Int. Conf. and Exhibition on the Practical Applications of Intelligent Agents and Multi-Agent Technology, UK, 1996, pp. 205-223.

9. GRUBER, T.R., **A Translation Approach to Portable Ontologies**, Knowledge Acquisition, 5(2), 1993, pp. 199-220.

10. KALFOGLOU, Y., SCHORLEMMER, M., **Ontology Mapping: the State of the Art**, The Knowledge Engineering Review, 18(1), 2003, pp. 1-31.

11. LEVENSHTEIN, I.V., **Binary Codes Capable of Correcting Deletions, Insertions, and Reversals**, Cybernetics and Control Theory, 10(8), 1966, pp. 707-710.

12. LI, J., LOM, **A Lexicon-based Ontology Mapping Tool**, Teknowledge Corporation, research report, 2004.

13. LUCK, M., Mc BURNEY, P., SHEHORY, O., WILLMOTT, S., **Agent Technology: Computing as Interaction – A Roadmap for Agent Based Computing**, AgentLink III, 2005.

14. Mc CALLUM, A., NIGAM, K., UNGAR, L.H., **Efficient Clustering of High-dimensional Data Sets with Application to Reference Matching**, Knowledge Discovery and Data Mining, 2000, pp. 169-178.

15. MILLER, G., BECKWITH, R., FELLBAUM, C., GROSS, D., MILLER, K., **Introduction to WordNet: An On-line Lexical Database**, International Journal of Lexicography, 3(4), 1990, pp. 235-244.

16. NOY, N.F., MUSEN, M.A., **The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping**, International Journal of Human-Computer Studies, 59, 2003, pp. 983-1024.

17. OPREA, M., **Mapping Ontologies in an Air Pollution Monitoring and Control Agent-Based System**, N. Lavrăc, L. Todorovski, and K.P. Jantke (Eds): DS 2006, LNAI 4265, Springer, 2006, pp. 342-346.

18. OPREA M., **A Case Study of Agent-Based Virtual Enterprise Modelling**, Proceedings of the 4th International Conference CEEMAS 2005, LNAI 3690, Springer, 2005, pp. 632-635.

19. OPREA, M., **Applications of Multi-Agent Systems**, Information Technology, R. Reis (Ed), Kluwer Academic Publishers, 2004.

20. PROTÉGÉ-2000: http://protégé.stanford.edu

21. SU, X., HAKKARAINEN, S., BRASETHVIK, T., **Semantic Enrichment for Improving Systems Interoperability**, Proceedings of the 2004 ACM Symposium on Applied Computing, ACM Press, Nicosia, Cyprus, 2004, pp. 1634-1641.

22. USCHOLD, M., KING, M., MORALEE, S., ZORGIOS, Y., **The Enterprise Ontology**, *The Knowledge Engineering Review*, 13(1), 1998, p. 31-89.

23. WORDNET: http://wordnet.princeton.edu/