19

ONTOLOGY-SERVICES AGENT TO HELP IN THE STRUCTURAL AND SEMANTIC HETEROGENEITY

Andreia Malucelli[1,2] and Eugénio Oliveira[1]
[1]*LIACC-NIAD&R, Faculty of Engineering, University of Porto,*
R. Dr. Roberto Frias, 4200-465 Porto, PORTUGAL
[2]*PUCPR, Pontificia Universidade Católica do Paraná*
R. Imaculada Conceição, 1155 - 80215-901 Curitiba PR, BRAZIL
malu@ppgia.pucpr.br, eco@fe.up.pt

In the Virtual Enterprises (VE) environment, interactions between distributed, heterogeneous computing entities representing different enterprises, people and resources, take place. These interactions, in order to be both syntactic and semantic compatible, need to follow appropriate standards (ontologies) well understood by all the participants. Evenfor each domain ontology, people may store their data in different structures and use different terms to represent the same concept. This paper focuses on ane effort to create an Ontology-Services Agent to monitor the communication acts taking place in a Multi-agent System. The Ontology-Services Agent provides help in solving the Structural and Semantic Heterogeneity problem, enabling appropriate conversations and making it possible meaningful agreements between agents representing different enterprises and resources in a VE environment.

1. INTRODUCTION

In a Virtual Enterprise (VE), a temporary consortium of partners and services is formed for specific purposes. These purposes could be a temporary special request, an ongoing goal to fulfill orders, or an attempt to take advantage of a new resource or market niche. The general rationale for forming the VE is to reduce costs and time to market while increasing flexibility and access to new markets and resources (Petrie and Bussler, 2003).

Software Agents and Multi-Agent Systems have been, for the last years, presented as a good paradigm for system architectures and, as a consequence, new agents' communication languages as well as appropriate platforms for agents' distribution have been released and experimented. Applications of such tools to Electronic Business domain brought up the need for the creation, representation and exploration of domain ontologies.

Even for each common domain ontology, people may store their data in different structures (structural heterogeneity) and use different terms to represent the same concept (semantic heterogeneity). Moreover there is no formal mapping between high-level ontologies.

This structural and semantic heterogeneity does not guarantee the consistency and the compatibility of the information present in the system and makes it much more difficult to establish a fruitful negotiation.

Our work is currently focused on the creation of an Ontology-Services Agent to monitor the communication acts taking place in Multi-agent System. The Ontology-Services Agent provides help in solving the structural and semantic heterogeneity problem, enabling appropriate conversations and making it possible meaningful agreements between agents representing different enterprises and resources in a VE environment. We here propose the use of intelligent agents and multi-agents technology as a framework for helping in the establishment of a Virtual Enterprise.

Ontologies and the structural and semantic heterogeneity problem are discussed in Section 2. Section 3 presents and explains the architecture proposed. A protocol for the Ontology-Services Agent Interaction Monitoring is presented in Section 4. Finally, Section 5 presents our current conclusions.

2. ONTOLOGIES AND THE STRUCTURAL AND SEMANTIC HETEROGENEITY PROBLEM

Ontologies were developed, in the field Artificial Intelligence, in order to facilitate knowledge sharing and reuse. Since the beginning of the nineties, ontologies have become a popular research topic investigated by several research communities, and the reason is in large due to promise a shared and common understanding of some domain that can be communicated between people and application systems (Fensel et al., 2001).

In a Collaborative Organization Environment (COE), including business transactions where all the partners, both sending and receiving messages have to agree, it is necessary to share common standards.

In all types of communication, the ability to share information is often hindered because the meaning of information can be drastically affected by context in which it is viewed and interpreted (Ciocoiu et al., 2001), and the ability to share information may be hard due to the impossibility to have a unique ontology for each application domain.

Usually, each application, and more specifically in the context of the work we are doing, each agent has its specific, private ontology and it may not fully understand other agent's ontology.

Even in similar domains there are both syntactic and semantic differences between ontologies and it becomes necessary to deal with these problems.

Different people have a different vision of the world and, consequently people or agents may use different terms for the same meaning or may use the same term to mean different things. The defined natural-language definitions associated with terms are either too ambiguous to make differences evident, or do not provide information to resolve those differences. Successful exchange of information means that the agents understand each other and meaning accuracy is guaranteed. The interoperability problem happens when we have heterogeneous and distributed systems.

In order to solve the interoperability problem (Malucelli and Oliveira, 2003), both structural and semantic heterogeneity have to be dealt with. (Wache et al., 2001) defines these heterogeneities as follows:

Structural heterogeneity: meaning that different information systems store data and concepts using different structural relationships.

Semantic heterogeneity: considers the contents of an information item and its intended meaning. There are three main causes for semantic heterogeneity:

i. *Confounding conflicts:* occur when information items seem to have the same meaning, but differ in reality.

ii. *Naming conflicts:* occur when naming schemes of information differs significantly.

iii. *Scaling conflicts:* occur when different reference systems are used to measure the same value. An example is the use of different currencies.

Figure 1 shows a simple example where we may observe, using UML schemes, the structural and semantic conflicts. Suppose we have Ontology A and Ontology B, with different views, both for the same domain of music compact disc selling. It is really complex to discover the correspondent items. The Ontology A, for example, has an **Audio Compact Disc** concept where one of the attributes is **publisher.** The Ontology B has the **Music CD** concept with **publishing House** attribute whose meaning is the same as the **publisher** attribute. Thus the relation *is composed by* between **Audio Compact Disc** and **Artist,** has the correspondent relation, *has-performer,* between **Music** and **Performer** in Ontology B.

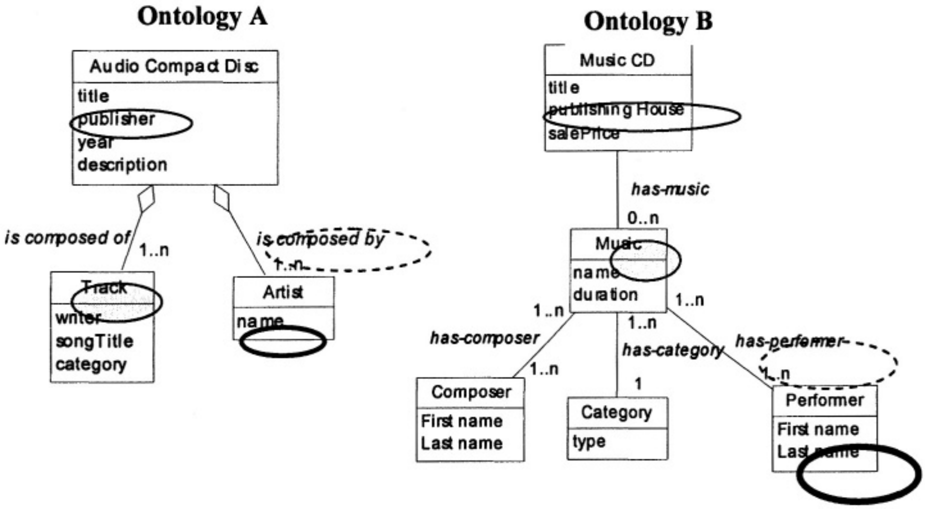

Figure 1 - Structural and Semantic Conflicts

Specifying a simple product like a music compact disc is relatively easy and there is a chance of always finding similarities in the description, but specifying a more complex product like a car or a plane may be really tough work.

The semantic and syntactic difference between ontologies is one of the most difficult problems and usually implies translation or mapping of ontologies.

It is important now to distinguish ontology translation from ontology mapping. Ontology translation is required when generating ontology extensions and querying through different ontologies. On the other hand, ontology mapping is the process of finding correspondence (mappings) between the concepts of two ontologies (Dou et al., 2003).

The mapping process is expressed by some rules, which express how concepts correspond to each other. When two concepts are correspondent, they mean the same thing. The mappings are generated either by ontology experts or by some automatic tool (Doan et al., 2002).

3. PROPOSED ARCHITECTURE

We are using intelligent agents and multi-agents technology as a framework for helping the communication and negotiation process for establishing a Virtual Enterprise. In our multi-agent system for VE formation, agents represent the enterprises and customers in the system. The VE life cycle is decomposed in four stages (Rocha and Oliveira, 1999):

1. *Identification of Needs:* appropriate description of the product or service to be delivered by the VE, which guides the conceptual design of the VE.
2. *Formation* (Partners Selection): automatic selection of the individual enterprises (partners) which, based in its specific knowledge, skills, resources, costs and availability, will integrate the VE.
3. *Operation:* Control and monitoring of the partners' activities including resolution of potential conflicts, and possible VE reconfiguration due to partial failures.
4. *Dissolution:* Breaking up the VE, distribution of the obtained profits and storage of relevant information for future use of the Electronic Institution,

Several problems are involved in the VE formation process, and one of great importance is the lack of understanding that may arise during agents' interaction due to the structural as well as semantic representation heterogeneity. In the *Identification of Needs* phase it is necessary to describe the needed product or service based on some ontology. For the *Partners Selection* phase, the knowledge, skills, resources, costs and availability have also to be specified in a way that it is understood by all the participants. Even for the *Operation* and *Dissolution* phases, a consistent and compatible communication is necessary.

The easier way of solving this problem is to have a common ontology available, which may be understood by all the enterprise delegate agents participating in the process. However, it is not sure that all the agents will use a common ontology. In our case we are using the multiple ontology approach (Wache et al., 2001), where each agent explores its own ontology. It is a decentralized and distributed approach according to our multi-agent system architecture.

An Ontology-Services Agent is proposed to be included in the framework for agents' interoperability, to monitor the communication taking place and help in the structural and semantic heterogeneity problem, just in time, without needing a previous and tedious complete ontology mapping process.

This framework includes 4 types of agents: facilitator agent, enterprise agents (good/product/services suppliers and customer), and ontology-services agent. The facilitator agent and enterprise agents - suppliers (SEAg) and customers (CEAg), are cooperating together through a website with the objective of providing or getting goods/products/services, in collaboration, but keeping their own preferences and objectives. An ontology-services agent is involved in all the process for monitoring and facilitating the communications and negotiations between agents.

The facilitator agent is the entity that matches the right agents and supports the negotiation process.

The enterprise (customer and suppliers) agents and ontology-services agent have to register themselves to be able to communicate. Each agent has its own private ontology, built in a private and unknown (to the overall system) process.

Customer Enterprise Agents represent enterprises interested in buying components to build a final product. Several suppliers in the world may have these components with different prices and conditions. Each CEAg sends a message (Identification of Needs) to the facilitator announcing which composed product/service is needed to configurate.

Supplier Enterprise Agents represent enterprises interested in providing some kind of product/service/good. Whenever a needed product, the facilitator agent conveys this announcement to all registered interested supplier enterprise agents.

Ontology-Services Agent keeps monitoring the whole conversation trying to help when some message is not fully understood by some of the participants.

Figure 2 shows an instance of the multi-agent system architecture for the VE formation process. Each Enterprise Agent (Supplier or Customer) has its own architecture and functionalities (some developer will design and build the ontology with some tool and, later, the agent will access the generated file/database).

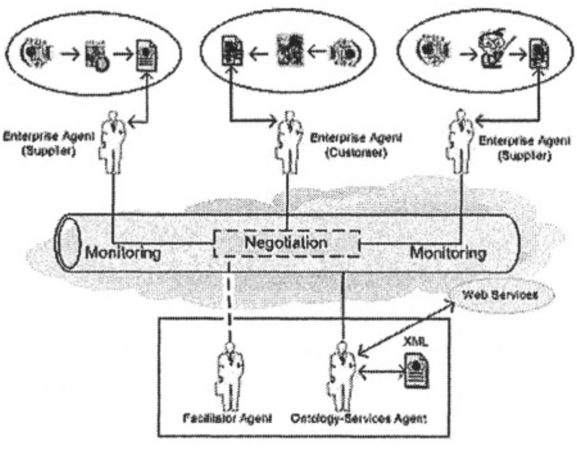

Figure 2 - System Architecture

The Enterprise Agents communicate with the Facilitator Agent whenever they have interest in buying or selling products/goods/services. The Ontology-Services Agent is monitoring the negotiation and communication, accessing a local ontology and web services whenever it is necessary, enabling to appropriate conversations and making it possible to reach agreements between agents representing different enterprises and resources in a VE environment.

4. ONTOLOGY-SERVICES AGENT MONITORING

We have created the ontology-services agent (OSAg) for trying to solve the problem, or part of the interoperability problem, in such a way that it is not necessary to translate or map all the ontologies involved.

The ontology-services agent monitors all the communication. When the Facilitator Agent sends an announcement asking for some product/service/good (the interaction indicated by number **1,** in Figure 3) required by Customer Enterprise Agent, all the Supplier Enterprise Agent may or may not understand the description of the product/service/good announced.

If the SEAg understand the description and if it is of their interest, they may formulate proposals (in Figure 3, the interaction number **2**). If one SEAg does not understand the message, it sends back another message with the content "unknown" (the interaction number **3,** in Figure 3). This may happen because either it may be using a different ontology or because it really does not have the requested product/service/good description.

If the answer is "unknown" the ontology-services agent who is monitoring all the messages, understands that this agent may have the component but he may not know the meaning of some term used in the message.

Figure 3 partially shows the protocol used for the messages exchanged between SEAg and OSAg. When SEAg sends a message whose contention "unknown", the OSAg starts the process of asking for more information. The OSAg keeps asking for information (Figure 3, the interaction number **4**) until it discovers (or gave up) the correspondent term in the SEAg ontology.

Examples of messages exchanged by ontology-services agent and supplier enterprise agent asking for information and another answering information, in KQML are described below:

```
ask-about
:sender ontology-services agent
:receiver supplier enterprise agent
:reply-with what-concept
:content (writer, songTitle, category)
```

The ask-about message may ask reply-with: what-concepts, what-attributes, what-relations, what-description.

```
reply
:sender supplier enterprise agent
:receiver ontology-services agent
:reply-to what-concept
:content (Track)
```

In the reply message, the answer may include the concept found for the correspondent given attributes or unknown, which means the correspondent attributes were not found.

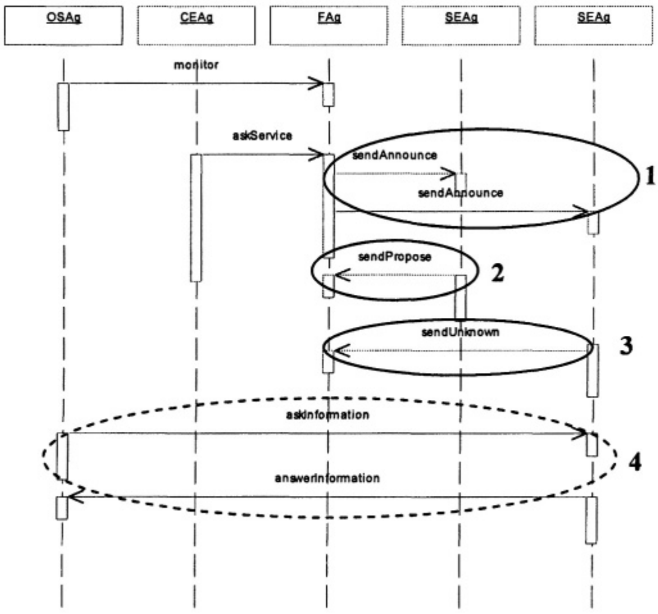

Figure 3 – Ontology-Services Agent Monitoring Protocol

Therefore, in order to help the resolution of the incompatibility, the ontology-services agent exchange messages with supplier enterprise agent asking for more information. The generic steps are below, where the ontology-services agent:

1. Searches for synonymous in its own ontology identifies an equivalent item although expressed in different terms.
2. Searches for characteristics (attributes) of that item, which may help the provided ontology to find another item with the same characteristics.
3. Searches for structural relations like "is", "is-a", "is-part-of", "composed-of", and "compose-by", in order to find out a different way of expressing the unknown term.
4. Asks for a description of the term to select the most representative words to be compared with the term concept description.
5. If in the end of all this process it was possible to find out some matching terms, a confirmation protocol is used to validate the term.
6. If the term is validated, then it is sent to SEAg that did not know the term, to make it possible to participate in the negotiation process.
7. The term is included in the agent services ontology, which can be used in the next negotiation round, avoiding having all the process repeated for this same situation once again.

5. CONCLUSION

The meaningful interaction between distributed, heterogeneous computing entities, in order to be both syntactic and semantic compatible, need to follow appropriated standards well understood by all the participants. Some standards are being developed, but what concerns ontologies, there is neither a standard ontology language nor a standard ontology knowledge representation.

This lack of standardization, which hampers communication and collaboration, is known as the interoperability problem (Willmott et al., 2001). Even in similar domains there is syntactic and semantic differences between ontologies and it is necessary to deal with these problems.

Several problems involved in the overcoming of syntactic and semantic heterogeneity are difficult to be solved, at least nowadays. However, on efforts have been directed towards finding possible ways to resolve, at least, parts of this complex problem.

We here have proposed the use of intelligent agents and multi-agents technology as a framework for helping the automatic establishment of a Virtual Enterprise. An Ontology-Services Agent was proposed to help in the communication and negotiation process between enterprise agents when interoperability problems happen. In an open environment where enterprise agents can register themselves and negotiate, ontology-services agent is used to monitor and ease the communication process at the moment when it is happening.

6. REFERENCES

1. Ciocoiu, M., Gruninger, M., Nau, D.S. Ontologies for Integrating Engineering Applications. Journal of Computing and Information Science in Engineering 1(1): 12-22. March 2001.
2. Doan, A., Madhavan, J., Domingos, P. Halevy, A. Learning to map between ontologies on the semantic web. In Proceedings of the World-Wide Web Conference (WWW-2002), 2002.
3. Dou, D., Mcdermott, D., Qi, P. Ontology Translation on the Semantic Web. In Proceedings of International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2003). LNCS 2888. Springer-Verlag. Berlin Heidelberg. pgs. 952-969. 2003.
4. Fensel, D. Horrocks, I., van Harmelen, F. McGuinness, D., Patel-Schneider, P. F., OIL: Ontology Infrastructure to Enable the Semantic Web. IEEE Inteligent Systems, 16(2), 2001.
5. Malucelli, A., Oliveira, E. Ontology-Services to Facilitate Agents' Interoperability. In Proceedings of the Sixth Pacific Rim International Workshop on Multi-Agents (PRIMA 2003). Eds. Jaeho Lee, Mike Barley, Springer-Verlag. LNAI 2891. pgs 170-181, Korea. Novembro 2003.
6. Petrie, C. Bussler, C. Service Agents and Virtual Enterprises: A Survey. IEEE Internet Computing. IEEE Computer Society. pgs 2-12. July/August 2003.
7. Rocha, AP., Oliveira E., An Electronic Market Architecture for the formation of Virtual Enterprises. In Infrastructures for Virtual Enterprises – Networking Industrial Enterprises, Ed. L.Camarinha-Matos e H.Afsarmanesh, Kluwer Academic Publishers, pgs. 421-432, 1999.
8. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information-A Survey of Existing Approaches. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01). Workshop: Ontologies and Information Sharing, Seattle, USA. 2001.
9. Willmott, S., Constantinescu. I, Calisti, M. Multilingual Agents: Ontologies, Languages and Abstractions. In Proceedings of the Workshop on Ontologies in Agent Systems. 5th International Conference on Autonomous Agents. Montreal. Canada. 2001.