

Implementing Explanation Ontology for Agent System

Xiaomeng Su¹, Mihhail Matskin², Jinghai Rao¹

¹*Department of Computer and Information Sciences,
Norwegian University of Science and Technology, 7491 Trondheim, Norway
Email: {xiaomeng, Jinghai}@idi.ntnu.no*

²*Department of Microelectronics and Information Technology,
Royal Institute of Technology, SE-164 40 Kista, Sweden
Email: misha@imit.kth.se*

Abstract

The overall issue addressed in this paper is to improve semantic interoperability among and across agent systems. We propose to use explanation as a way to approach that aim. The explanation process is expressed in terms of an explanation ontology shared by the agents who participate to the explanation session. The explanation ontology is defined in a way general enough to support a variety of explanation mechanisms. The paper describes the explanation ontology and provides a working through example illustrating how the proposed generic ontology can be used to develop specific explanation mechanism. Finally, the ontology is being integrated into a running agent platform – Agora to demonstrate the practical usefulness of the approach.

1. Introduction

Over the past few years, researchers and industry have both been involved in a great drive towards enabling interoperability between diverse information sources through the explicit modeling of concepts used in communication. Semantic web is one of the most significant endeavors to that end. The hope is that the semantic web can let computers process the interchanged data in a more intelligent way.

Ontology is a key factor for enabling interoperability in the semantic web [2]. One of the most cited definitions of ontology is that “An ontology is a formal, explicit specification of a shared conceptualization” [8]. Ontologies are central to the semantic web because they allow applications to agree on the terms that they use when communicating. Within a multi-agent system, agents represent their “view of the world” by explicitly defined ontologies. The interoperability of such a multi-agent system is achieved through the reconciliation of these views by a commitment to common ontologies that permit agents to interoperate and cooperate. Thus,

ontologies, if shared among stakeholders, will improve system interoperability across agent systems.

However, there has long been argued that there is and will be no one single universal shared ontology, which will be applauded by all players. Therefore, it is highly likely that the involved agent systems may use different ontologies to represent their view of the domain. The issue of improving system interoperability will therefore, boils down to the reconciliation of different ontologies used in different agent systems.

This reconciliation usually exists in the form of mappings that relate concepts in the two ontologies. The mappings are normally computed off-line, either manually or automatically (in most case, semi-automatically). There have been a number of approaches for deriving mappings for ontologies [9][10][11]. This paper is yet not another endeavor to propose new approaches for mapping generation, rather, it concentrates on how to explore the mapping derived by other methods and incorporate them into agent systems in order to achieve the goal of greater semantic interoperability within and across agent systems.

We base our research on an agent platform, Agora. The Agora system is a multi-agent system environment, which provides support for cooperation between agents [6][7]. Ontologies are used to give semantic meaning to the contents of messages sent between agents.

Explanation, like any other kind of agent communication, is a complex task and a range of agreements has to be made before any meaningful communication could ever happen. Therefore, in order to use explanation for reaching consensus on terms used in heterogeneous multi-agent systems, we first need the agreement to use a consensual terminology for enabling the explanation process. We call the agreed conceptualization of the explanation process explanation ontology. Committing to the explanation ontology is the prerequisite for successful explanation.

The rest of the paper is organized as follows. First we present the proposed explanation ontology, which includes the interaction protocol, the explanation profile

and the explanation strategy. Next, an example is provided to illustrate the idea. Then, we demonstrate how it can be incorporated into Agora. Related work and future study conclude the paper.

2. The Explanation Ontology

We assume that agents engaged into an explanation process will have two types of knowledge:

- Knowledge concerning the domain of interest, i.e. the concepts, which represent the objects that are to be explained and the parameters of explanation.
- Knowledge concerning the explanation, i.e. what are the concepts that describe the explanation process and what are the permitted interactions in the process.

It is the second part, i.e. the knowledge concerning the explanation, which constitutes the explanation ontology. The structuring of the ontology is needed to provide three types of knowledge about the explanation. They are knowledge about interaction (explanation interaction protocol), knowledge about description of explanation (explanation profile) and knowledge about how explanations are derived (explanation strategy). Each of the three provides an essential type of information about the explanation, as characterized in the rest of the paper.

The generic explanation ontology is intended to capture similarities between different explanation mechanisms. It can be used as classification framework that permits the analysis of the explanation mechanisms available, and more important to develop new ones. Furthermore, by committing to the same high-level concepts, the communication among agents is facilitated, in a more flexible way.

It should be noted that while we define a particular generic ontology, the construction of alternative approaches is allowed. Our intention here is not to prescribe a single approach in each of the three areas, but rather to provide generic approaches that will be useful for the majority of cases. In the following three sub sections we discuss the resulting explanation interaction protocol, explanation profile, and explanation strategy in greater detail.

2.1. Explanation interaction protocol

An explanation interaction protocol defines how the explanation is performed. In particular, it describes the dataflow and possible interactions among participants during the explanation process.

A concept explanation interaction protocol is a pre-specified pattern of message exchange between agents. It is a pragmatic solution for agent conversation, so that an

agent can engage in meaningful conversation with other agents, simply by carefully following the interaction protocol. A general explanation interaction protocol is built based on generalizing the commonalities of different protocols. Figure 1 identifies the main concepts of the general protocol. There are several roles involved in the protocol. Each role is played by one or more agents (participants). Possible involved roles are Initiator, the agent who asks for explanation; Explainer, the agent who provides explanation; and Explanation Manager, who mediates the explanation. A protocol is also guided by a number of explanation rules. Each rule tells what action for a role to take when a certain precondition is met.

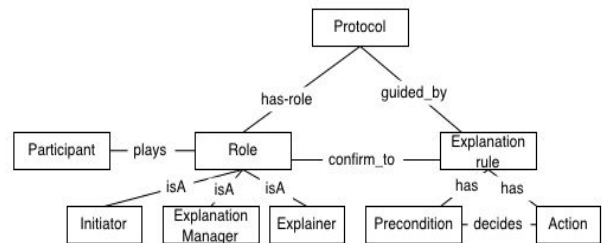


Figure 1: An ER model of the general explanation interaction protocol

By refining the concepts in the general protocol we can define different specific protocols. The refinement of a concept is achieved by restricting the value set of an attribute or by adding new attribute to concepts. For example, we define the concept protocol has an attribute `has_role`, whose minimum cardinality is 2. By that, we say at least two agents (initiator and explainer) need to be engaged into a protocol. However, when defining a mediated explanation protocol, we restrict the minimum cardinality to 3, since an explanation manager, who functions as a mediator, is added into the interaction. We consider a general framework for presenting such protocols as well as some of the specific protocols. A specific explanation interaction protocol is presented in the example section.

2.2. Explanation profile

The explanation profile defines the main concepts that are used in the explanation. Figure 2 is an ER (Entity Relation) model of the concepts.

The main concepts in the explanation are query and explanation. Each query is in correspondence with a number of explanations. The query consists of one source element, one source ontology and one target ontology. Each explanation concerns two ontology elements, one source and one target. Each ontology element belongs to an ontology. An explanation also has a type, which

defines the kind of relationship between the corresponding source and target ontology elements. Ideally, the correspondence between ontology elements should be translation where semantics of the concept are completely preserved. However, transformations (mappings that lose some of the semantics) are also permitted to allow for approximate explanation. Thus the type of an explanation can be one of the following:

- Similar concept (car \approx automobile)
- Narrower concept (station wagon is a car)
- Broader concept (car is-a-kind-of vehicle)
- Related concept (car is related with transport)

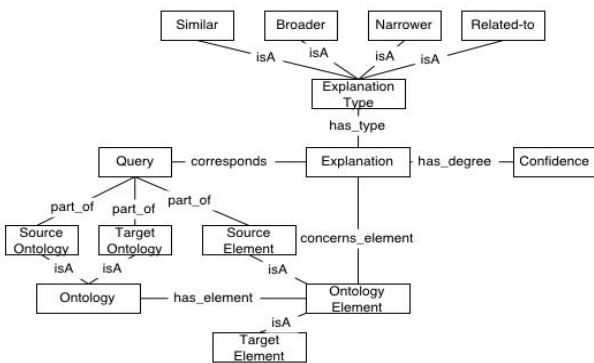


Figure 2: An ER model of the main concepts in the explanation profile

A degree is associated with each explanation to suggest the confidence of the mapping. Additional explanation method can be easily added, by making another instance of **ExplanationType**, for example, logical expression.

2.3. Explanation strategy

The explanation strategy describes how the explainer does the analysis of discrepancy and what are the evaluation criteria for the initiator to decide whether accept a certain explanation or not.

For the explainer, different strategies may be configured by using different combinations of the following aspects:

- Source of explanation. The explainer may use different information sources to derive explanation. Possible sources are mappings between local ontologies, mappings between local and global ontologies, and external lexicon (e.g. WordNet)
- Ranking strategy. When multiple explanations are available for a given concept, a ranking strategy is need to determine which one to use in the first place. The ranking strategy may take the form of a set of ranking rules or to employ a ranking function

to calculate a numerical figure for each explanation in the result set.

- Termination criteria. The explainer could terminate the process, when no more explanation is available, or when a predefined maximum round of explanation is exceeded.

On the initiator side, two main aspects define its strategy:

- Acceptance criteria. This is the strategy about when to accept an explanation, when to reject and when to ask for more explanation.
- Termination criteria. It defines when to terminate the explanation process.

Based on the general framework discussed above, specific explanation strategy can be introduced (for example, the explanation strategy in next section.)

3. A working through example.

In order to better illustrate the idea of how to make use of the generic explanation ontology presented above to develop new explanation mechanism, we have considered the application scenario of an electronic market place. In an electronic market place, where buyers and sellers are brought together, each individual participant, (possibly software agent) may use its own product catalogue to represent the required products respectively[3]. In that context, making it possible for agents to “understand” what is required and what is offered becomes non-trivial. We see explanation as one way of solutions.

3.1. Two product catalogues.

Let us look at two product catalogues in Figure 3a,b.

Segment	Family	Class	Commodity	Title
42	00	00	00	Medical Equipment and Accessories and
43	00	00	00	Information Technology
	Family	Class	Commodity	Title
	19	00	00	Communications Devices and Accessori
	19	15	00	Personal communication devices
	19	15	01	Mobile phones
	19	15	02	Pagers
	19	15	03	Pay phones
	19	15	04	Phones

Figure 3a. Segment of UNSPSC

Classification	Description
24	Communication technology, office -
24-01	Hardware (inform. technology)
24-01-03	Hardware (workstation)
24-01-03-01 [BSA]	Hardware (workstation)
24-01-03-xx [BSA]	Hardware (workstation) (discontinued)

Figure 3b. Segment of Eclass

We assume that a supplier uses the UNSPSC¹ standard, while a buyer uses the Eclass² standard. We further assume that the mappings between two standards are maintained by a service agent. How these mappings are derived is beyond the scope of this paper and articles like [9][10][11] can be referred to. A general mapping process typically involves analyzing the ontologies and comparing them to determine the correspondence among concepts and detect possible conflicts. A set of mapping assertions is the main output of a mapping process. A couple of typical mapping assertions are presented in Table 1.

Table 1: Example mappings between two catalogues

UNSPSC	Eclass	Type	Confidence
43211507 Workstation or personal computers	240103 Hardware (workstation)	Similar	.95
43211901 Monitors	240106 Screen	Similar	.90

3.2. Specific explanation interaction protocol

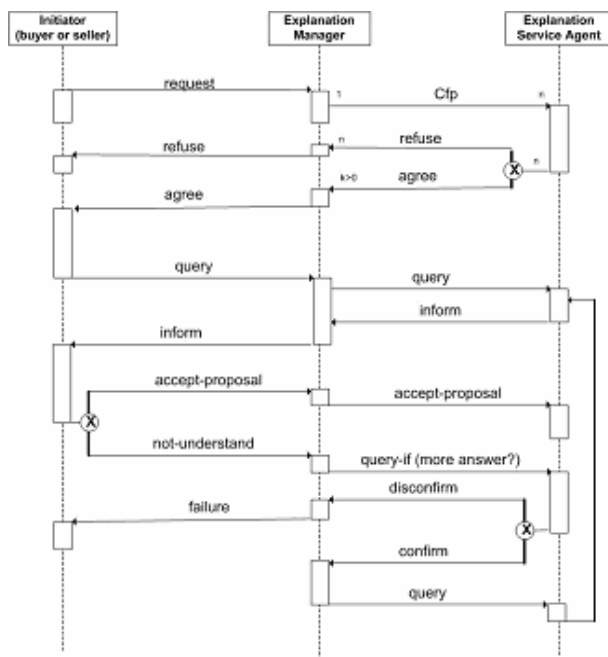


Figure 4: Explanation interaction protocol

As illustrated in section 2.1, a specific explanation interaction protocol can be introduced by refining and instantiating the generic protocol. Figure 4 presents one of such specific protocols using AUML[12]. This

protocol uses the FIPA[5] reserved communicative acts, thus allow maximum interoperability with other FIPA compliant agent system.

This protocol has three roles, among which the Explanation Manager acts as a mediator between Initiator and Explainers. The rules that govern the process of the interaction are:

1. The Initiator requires help from the explanation manager.
2. The Explanation Manager acts as an auctioneer to choose one or more Explainers to provide explanation service.
3. During explanation, the Explanation Manager acts as a message mediator between the Initiator and the Explainers.
4. After getting query the Explanation Manager tries to consult the Explainer to produce an explanation. The Explainers can explain the terms by different strategies presented in section 3.3. If no explanation can be produced then a failure message is sent out.
5. Upon receiving the explanation, the Initiator examines the explanation and replies whether it is satisfied with the proposed explanation (the accept message) or not (the not-understand message).

The rules prescribed above are formally represented by using preconditions and actions. Each precondition is uniquely identified by the performatives in this explanation interaction protocol. Each performative has unique meaning in the context of explanation ontology (Table 2). They may have other meaning in other ontologies. A combination of performative and a specific ontology indicate a unique intended act in the scope of the agent system.

In this implementation, the Explanation Manager Agent manages the interaction protocol in terms of initializing and terminating the process as well as mediating messages between the Initiator and Explainer. This solution allows simultaneous running several interaction protocols for the same concept explanation or performs one after another (if the previous protocol doesn't satisfy the Initiator).

Table 2 Meaning of performatives

Performative	Implemented Meaning
request	Initiator requests explanation service between two ontologies.
cfp	Explanation Manager invites Service Agents to provide explanation service
refuse	The service agent refuses to provide explanation service
agree	The service agent agrees to provide explanation service

¹ www.unspsc.org

² www.eclass.de

Query	The agent asks for explanation of a particular concept.
Failure	The Explanation Manager can't give meaningful explanation
Inform	The agent informs about the explanation
accept-proposal	The explanation is accepted
not-understand	The initiator can't understand the explanation
query-if	The Manager queries the service agents whether they can provide more explanation
Disconfirm	The explainer can't provide more explanation
Confirm	The explainer can provide more explanation

3.3. Specific explanation profile and strategy

The generic explanation profile is used for queries and explanations that are passed between agents.

Based on the general framework in section 2.3, a specific explanation strategy is developed. For the Explainer, a hybrid approach for explanation derivation is proposed. The hybrid approach contains local ontology alignment, global ontology alignment and term alignment. They can be employed simultaneously or one after another, according to the resources available.

Local ontology alignment refers to the situation, where both the source and target ontology are known and mappings between the two ontologies are used as source for explanation. The mappings are computed off-line and explored by agents during runtime.

In the example we proposed in section 3.1, the mappings between the two standards are represented (encoded) by committing to the explanation profile in section 2.2. An example representation is like the following, which says that element 43211507 (workstation or personal computers) in UNSPSC is similar to element 240103 (Hardware workstation) in Eclass and their similarity degree is 0.95.

```
<explanation rdf:ID = "explanation-00" >
  <source rdf:resource = "UNSPSC:43211507">
    <sourceOntology rdf:resource =
      "http://www.unspsc.org" />
  </source>
  <target rdf:resource = "Eclass:240103">
    <targetOntology rdf:resource =
      "http://www.elcass.de" />
  </target>
  <type rdf:resource = "similar" />
  <confidence rdf:resource = "0.9" />
</explanation>
```

Global alignment considers the case that the mappings between the source ontology and the target ontology are

not available, but both the initiator and explainer know a third-party ontology so that all terms in the source ontology and target ontology can be explained by the third-party ontology. In this example, the third party ontology is NAICS³ (North American Industry Classification Systems). We assume that mappings between NAICS and the initiator's local ontology (EClass) and mappings between NAICS and the source ontology (UNSPSC) can be accessed by the initiator and the explainer respectively. A NAICS term "display terminals" (33411131005) which has a mapping both to "monitor" and "screen" can be used as an explanation.

Term alignment intends to use linguistically related terms as a source for explanation. It is employed when no specific mappings of the two involved ontology are known in prior. A general-purpose lexical database WordNet is used as external resources to derive synonyms (similar concepts), hypernyms (broader concepts) and hyponyms (narrower concepts) of the query concept. The ranking strategy is made up of a set of rules as well. High ranked term is chosen and wrapped up into message according to the explanation ontology.

For example, "monitor" has a direct synonym "display" in WordNet, so the explainer who offers term alignment service gives "display" as the first explanation. However, "display" is not in the vocabulary of EClass, thus the Initiator can't understand it either. Then the explainer provides a new explanation "screen" which is hyponym of "display" and at the same sense of "monitor". The Initiator understands "screen" so that the explanation is accepted.

On the initiator side, the decision on whether to accept an explanation or not is based on the notion of "understanding". The idea is based on the fact that the suggested concept exists in its vocabulary set (ontology) and the confidence level associated with that explanation is satisfactory. Otherwise, the agent may ask for another round of explanation or give up the process. Agent may pre-define a maximum threshold to restrain the number of rounds that can be performed.

4. Implementation

4.1. The Agora system

The Agora system is a multi-agent environment, which provides supports for cooperative work between agents. The concept of cooperative node [7] is an infrastructure element where agents participating in a cooperative activity can register themselves and get support for communication, coordination and

³ <http://www.census.gov/epcd/www/naics.html>

negotiation. Basic functionality of the Agora node is presented in Figure 5.

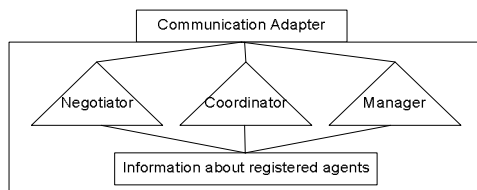


Figure 5. Agora node functionality

Each Agora node has a Communication Adapter, Manager Agent, Negotiation and Coordination Agents. The Communication Adapter allows agents communicate in Agent Communication Language, which is compliant with both FIPA ACL[5] and KQML [4]. Complexity of the Manager Agent can be different for different applications, however, basic matchmaking functionality is provided by default in all Managers. The default matchmaker can be overridden in order to support more advance functionality, such as:

- Matchmaking using ontology and semantic relations.
- Event handling.
- Decision making.
- Processing queries about registered activities, ontology used and other general information
- Handling agent registration/unregistration protocols
- Pro-active reasoning with available knowledge
- History maintenance and analysis

Negotiator and Coordinator are agents, which manage, correspondingly, negotiation and coordination protocols. In particularly, they can implement functionality of auctioneers in auctions or managers in Contract Net Protocol.

Agents in the system are generated by an agent deployment block and they have a linear Planner, Knowledge Base and Executor Module. The default agent implementation can be overridden by a user via attaching specific functional components to the agent deployment block (for example, by attaching a proprietary planner). However, if there is no need in modifying the default agent functionality then a simple agent generation process can be applied (see Figure 6).

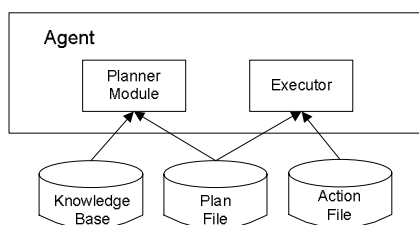


Figure 6. Simple agent architecture

For more details about the Agora system we refer to [6][7].

4.2. Implementing explanation ontology in Agora

In order to support the explanation function, we need to integrate the explanation ontology into the Agora architecture. The following requirements should be met:

1. Agents should be registered at some Agora node to get platform support.
2. The interaction protocol needs to be implemented.
3. The explanation profile needs to be incorporated into the agent message content.
4. The explanation strategy should be implemented by the involved agents.

The first requirement can easily be satisfied. When an explanation process is triggered, a new Agora node (Explanation Agora) is created, where Initiator, Explainer and Negotiation agents are registered (Negotiation agent plays a role of the Manager Agent described in the Section 2).

The second one is met by defining the interaction protocol using Agora plan file. The plan-file can be written manually (by the user) or generated automatically by the agent planner. Upon registration, Initiator, Explainer and Manager all import their plan and action files. Agent interaction will then be executed under the guidance of the plan/action file.

The third requirement is managed by encoding the explanation profile in the “content” part of the message. The explanation profile here can be viewed as a type system for the “content”. Within the “content” is a RDF statement.

The fourth requirement is met by defining appropriate action files for the involved agents. For service agents, a strategy on how to derive explanation is implemented either directly by the action file or by writing a code URI in the action file. That URI points to a computation component, which in our case is a Java class. The code URI solution is used only when the action file is not enough to express the algorithm. Depending on whether the service agent performs ontology alignment or term alignment, it uses different derivation strategy as is discussed in section 2.3. The set of rules that define rankings of result sets are instantiated by interacting with the user, which the agent is working on behalf of. For initiator, a strategy on evaluating the proposed explanation is implemented by a Java component. That component has access to the agent’s own ontology collection and is capable of locating specific concepts in the ontology. The minimum threshold for confidence level is set by the user and can be tuned if necessary.

5. Related works

In the research area of knowledge engineering, a number of ontology integration methods and tools exist. Prototyping tools for ontology merging and alignment have been developed, like the PROMPT system [13], and the Chimaera tool [14]. Most of the focus in these tools has been to support human experts -- not computers -- in detecting mismatch, suggesting solutions, and cooperating work during the reconciliation process. In our approach, however, focus is to meet understanding gaps dynamically for agents. Ontology aligning and ontology explanation are in fact complimentary, in the sense that ontology explanation provides a method to use the mappings derived by the aligning tools to achieve better understanding between agents.

Bailin and Tuzskowski introduced the idea ontology negotiation in their work [1]. They developed an ontology negotiation protocol in the form of application program interface (API), which enables agents to cooperate in performing a task, even if they are based on different ontology. Their work shares similar intention with ours but differs in the proposed solution. One of the major differences between stems from the fact that we consider generic framework for the explanation process as well as specific explanation mechanisms that commit to the generic framework. Furthermore, we also think of integrating and implementing the explanation ontology into the Agora system.

6. Conclusion and future works

In this paper, we present how to use incremental explanation as a way for two agents, who use different ontologies, to come to a partial understanding.

We present a generic explanation ontology, which provides three types of essential knowledge about the explanation. They are explanation interaction protocol, explanation profile and explanation strategy. The generic ontology can be used to develop specific explanation solutions as illustrated by a working through example. It could also be used as a classification framework to compare different explanation mechanisms.

The problem of semantic interoperability is hard and our research question is how much we may achieve of mutual understanding by, e.g. explanation. To examine our solution, we integrate the approach into an agent system – Agora. This also provides an anchor point for accommodating potential useful semantic reconciling techniques into agent systems.

For the generic explanation ontology we proposed to be really useful in a wider context, the quality of the ontology, i.e. how complete and valid it is, need to be

further investigated. Another aspect we are planning to investigate is to develop more specific explanation mechanisms to test the flexibility of the generic ontology.

Acknowledgements

This work is supported by Accenture Norway and the Norwegian Research Foundation in the framework of the Information and Communication Technology (IKT-2010) program - the ADIS project.

References:

- [1] Bailin, S. and Truszkowski, W.: Ontology negotiation between scientific archives. In *Proceeding of the 13th International Conference on Scientific and Statistical Database Management*, IEEE Press, 2001
- [2] Berners-Lee, T., Hendler, J., and Lassila, O.: The semantic web, *Scientific American*, May 2001,
- [3] Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A.: *Procuat Data Integration for B2B E-Commerce*. *IEEE Intelligent Systems* 16, 2001.
- [4] Finin, T., Labrou, Y. and Mayfield, J.: KQML as an Agent Communication Language. In J. M. Bradshaw (ed.), *Software Agents*. AAAI Press 1997.
- [5] Foundation for Intelligent and Physical Agents (FIPA). URL: <http://www.fipa.org/repository/bysubject.html>.
- [6] Matskin, M., Kirkeluten, O. J., Krossnes, S. B. and Sæle, Ø.: Agora: An Infrastructure for Cooperative Work Support. In *Multi-Agent Systems*. T. Wagner, O. Rana (eds.) *Infrastructure for Agents, Multi-Agents, and Scalable Multi-Agent Systems*. Springer Verlag, LNCS 1887, 2001.
- [7] Matskin, M.: Multi-Agent Support for Modeling Co-operative Work. In T. Yongchareon, F. A. Aagesen, V. Wuwongse (Eds.) *Intelligence in Networks*. The Fifth International Conference SMARTNET'99, Thailand, Kluwer Academic Publishers, 1999.
- [8] Gruber T. R.: Translation approach to portable ontology specification, *Knowledge Acquisition*, 2(5) 1993.
- [9] Omelayenko B.: Integrating Vocabularies: Discovering and Representing Vocabulary Maps, In *Proceeding of the first semantic web conference (ISWC-2002)*, Springer-Verlag LNCS 2342, 2002.
- [10] Bergamaschi S., Guerra F. and Vincini M.: A Data Integration Framework for e-Commerce Product Classification, In *Proceeding of the first semantic web conference*, Springer-Verlag LNCS 2342, 2002.
- [11] Su X.: Ontology mapping through analysis of model extension, In *proceeding of the CAiSE Forum 2003*.
- [12] Odell, J. J., Parunak, H. van Dyke and Bauer, B.: Representing agent interaction protocols in UML, *Agent oriented software engineering*, Springer LNCS 1957, 2001.
- [13] Noy, N. and Musen, M. A.: PROMPT: algorithm and tool for automated ontology merging and alignment, In *Proceeding of AAAI 2000*.
- [14] McGuinness, D., Fikes, R., Rice, J. and Wilder, S.: An environment for merging and testing large ontologies, In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, 2000.