Ontology–Related Services in Agent-Based Distributed Information Infrastructures

Ludger van Elst and Andreas Abecker
German Research Center for Artificial Intelligence (DFKI)
– Knowledge Management Group –
P.O. Box 2080, D-67608 Kaiserslautern, Germany
Phone: +49 631 205 3474, Fax: +49 631 205 3210
e-Mail (elst,aabecker)@dfki.de

Abstract

Ontologies are an emerging paradigm to support declarativity, interoperability, and intelligent services in many areas, such as Agent–based Computation, Distributed Information Systems, and Expert Systems. In the context of designing a scalable, agent-based middleware for the realization of distributed Organizational Memories (OM), we examine the question what ontology–related services must be provided as middleware components. To this end, we discuss three basic dimensions of information that have fundamental impact on the usefulness of ontologies for OMs, namely formality, stability, and sharing scope of information. A short discussion of techniques which are suited to find a balance in each of these dimensions leads to a characterization of roles of ontology–related actors in the OM scenario. We describe the several roles with respect to their goals, knowledge, competencies, rights, and obligations. These actor classes and the related competencies are candidates to define agent types, speech acts, and standard services in the envisioned OM middleware.

1. Introduction: Ontologies for Distributed Organizational Memories

In Knowledge Management (KM), it is widely accepted that ontologies as explicit specifications of conceptualizations [11, 12, 13] provide a useful means to facilitate access and reuse of knowledge. Typical utilization scenarios comprise discussion groups, search engines, information filtering, access to non–textual information objects, and expert–user communication [15]. In these applications ontologies serve as “specifications of discourse in the form of a shared vocabulary” [15]. This “shared understanding” seems to be particularly important for KM which typically deals with multi–actor scenarios. The vision of knowledge management assumes the comprehensive use of an enterprise’s knowledge, whoever acquired it, wherever it is stored and however it is formulated in particular. Organizational Memories (OM) [3] are proposed to support this vision by accumulating, structuring, and fostering utilization of *explicit knowledge*. In OMs ontologies can provide a vocabulary for *specifying information needs* as well as *information resources*.

Technical support for Organizational Memories is often based on *centralized approaches* which seem well–suited to guarantee that the *complete* information available is considered [5]. For instance, in the KnowMore framework (see Figure 1, [1]), the problem of several heterogenous information sources is tackled by the introduction of a *uniform knowledge description level*: The various information items are annotated by knowledge descriptions which are based on an agreed upon vocabulary, namely the information, enterprise, and domain ontologies. Hence, a centralized view upon a distributed information landscape is built. This centralized view is utilized by information processing and retrieval mechanisms to proactively deliver relevant information in the current process context at the application layer.

In the FRODO project [2] we extend the centralized KnowMore framework towards a distributed OM scenario. This is motivated by the drawbacks of centralized models:

- *They neglect the advantages of the distributed nature of knowledge (e.g. with respect to development and use) in an enterprise:* It is very expensive or even impossible to obtain a globally negotiated vocabulary. OMs could benefit from balancing both *local expertise* — which may result in not globally shareable knowledge — and *overall views* on higher levels.

- *Centralized approaches are cumbersome in changing environments:* An OM’s environment may for example

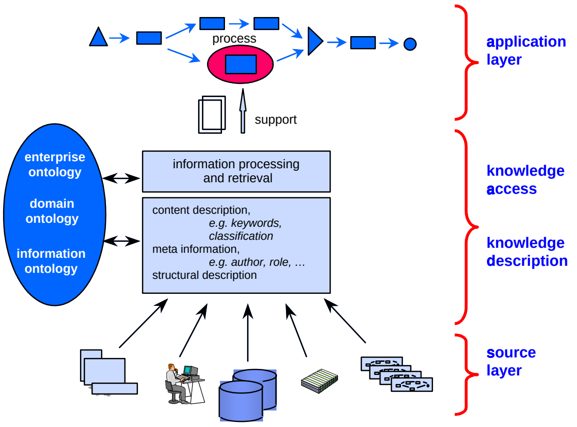

Figure 1. The KnowMore OM framework [1]

change due to reorganizations of an enterprise's structure. Furthermore, OM system are typically not established at once for a whole company, but introduced step by step (in terms of time and space) in various places (e.g., departments). To allow for a comprehensive management of knowledge, these OMs have to cooperate or to be integrated.

Consequently we propose a flexible, scalable framework for OM technology which supports two types of scalability:

1. **Vertical scalability** describes the ability of evolutionary growth *within one* OM:
   - integration of new services in order to meet additional requirements on the application level
   - incorporation of additional legacy systems in the source layer
   - extension / change of underlying ontologies for the knowledge description layer
2. **Horizontal scalability** means the cooperation *between several*, independently introduced OMs within the enterprise. For example, there may exist separate OMs for different departments of an enterprise (design, production, customer relationship mgt, etc.). In order to "globally optimize" the whole product lifecycle, information has to be used across the departments.
   Therefore, communication means *between* several OMs are needed. In such a horizontal integration scenario, communication and cooperation must be possible between all system layers of different OMs: cooperative information gathering or intelligent information integration are examples for a horizontal OM integration on the knowledge access layer. Cross–organizational workflows are an approach for facilitating integration on the application level.

These two types of cooperation between different OMs allow to see a "snapshot of an actual usecase" as an ad hoc configuration of a virtual OM, similar to a view in database systems. As a consequence of the considerations so far, we see that we examine an information landscape with various actors on the information provider and the information consumer side, as well as mediating information agents. Thus we have a strong need to establish a shared understanding between these actors.

The remainder of this paper is organized as follows: In section 2 we encourage a comprehensive view on three dimensions of information that have fundamental impact on the usefulness of ontology–based KM systems, namely *formality*, *stability*, and *sharing scope*. These considerations have their origin in an examination of real-world OM scenarios, particularly the need to balance cost and benefit and to make systems scalable and maintainable. A role model for ontology–related agents in OMs based on these considerations is proposed in section 3. [19] suggest such a role–oriented analysis as a natural step in their Gaia methodology for agent–oriented design, especially when it is manifest to take an organizational view on the application scenario. Section 4 shows how these roles can be embodied in the FRODO framework for distributed organizational memories. We conclude the paper by sketching an application example that will be used to show the suitability of our approach (section 5) .

2. Characterization of Information in the OM

Coming from the definition of ontologies, we identify important dimensions of information in OMs, analyze their interactions and sketch techniques that can be used to cope with the complexity resulting from these interactions.

2.1. Dimensions

Information in KM information systems can be described with respect to various dimensions, e.g., granularity, trustworthiness, or explicitness. When designing such a system, three dimensions are especially relevant:

- *Stability:* Information can have different levels of stability. For example, the contact person for a specific concern in an enterprise is normally quite permanent. However, if the enterprise sources out some functions to a call center, the knowledge about a contact person might become rather momentary, because each time one calls a new contact person is assigned.
- *Sharing Scope:* Information can be shared within a group or individual: individual because it has not yet been published or disseminated, or because it is aimed to be individual. Some notes on a post–it are

individual, whereas a design team should develop a more shared understanding of the product to be developed. Furthermore, knowledge and information can be shared within even larger structures, like a group of groups (e.g., across the whole enterprise with its various divisions).

- *Degree of Formality:* Information can be highly formalized (e.g., formal business process models or rules in an expert system), or it can be more informal (e.g., text documents). Formal information is meant to be machine-readable [18] and machine-interpretable.

It is desirable to identify information at the high end of these dimensions: Stable information can be reused over time; widely shared information can be reused across the organizational structure; formal information is a basis for powerful automation services. Several techniques have been developed to enable transitions along these dimensions (see below). In order to utilize them optimally in a comprehensive Knowledge Management system, a detailed understanding of their interactions is required.

2.2. Interactions

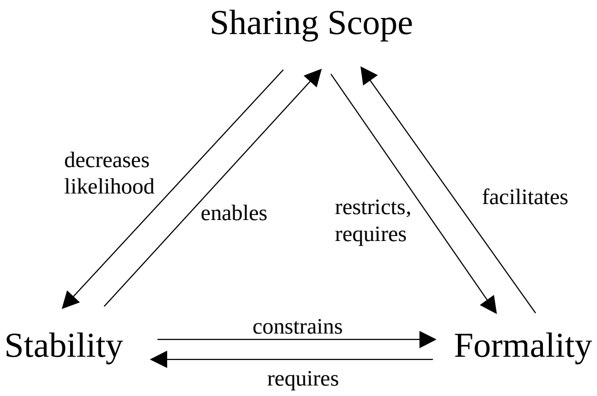

Figure 2. Sharing scope, stability, and formality of information in an OM

Below we describe how either two of the dimensions — stability, formality, sharing scope — mutually influence one another (cf. Figure 2):

- *Formality* vs. *Stability*: Reaching a high level of formality requires high effort: Specially trained personnel is needed; the process of formalization itself is time consuming, ambiguous and error–prone. This leads to high costs which normally only charge off if the application period of the formalization is orders of magnitude longer than the creation duration. However, the stability of a domain to be formalized is prevalently an externally determined factor. Therefore formalization degree must be chosen carefully, according to the expected stability. In the information delivery architecture described in [1], for instance, three ontologies are used for information modeling, namely: (i) the information ontology, a meta–model for structural description of information items (formats, types etc.), (ii) the enterprise ontology to specify creation and intended application context, and (iii) the domain ontology for content descriptions. Apart from the principal / ontological distinction, these ontologies differ especially in stability. While the vocabulary to structurally describe information (What essentially constitutes a book or a paper?) is quite stable, or at least expanding relatively monotonously (in contrast to the eighties, we now have web–documents), the domain ontology is often more a living organism: New topics become important, knowledge is acquired permanently, and respective conceptualizations are added or changed; old conceptualizations may be no longer valid, etc. The stability of an enterprise ontology resides between these two extremes. Enterprises change their organization from time to time. These changes may be slightly, within a fixed top–level ontology (What is a department, an employee, a process?), or they may be more rigorous, with a deep restructuring, e.g., as result of a merger. Consequently, for domain ontologies predominantly relatively weak formalizations (thesauri, concept lists, or hierarchies) are applied which can be maintained easily, often with automatic support. On the other hand, enterprise and information ontologies can be specified in a logical language with rich expressive power (e.g., TOVE [9]). In general, a detailed analysis of an expected ontology life cycle can be a powerful guide to achieve an optimal level of formalization in terms of costs and benefits.

- *Stability* vs. *Sharing Scope*: These two dimensions often tend to have a tradeoff. The more agents share a conceptualization, the more likely it is that some of them will break the commitments forming the basis of an ontology. Conversely, if a conceptualization is shared only between a couple of agents, stronger changes in the environment are needed to enforce a shift in the ontology. Whenever a large sharing scope is needed, it has to be taken into account that most negotiation processes (which facilitate sharing) are time-consuming. Therefore, high stability of the resulting ontology is desirable. Sometimes this stability is artificially achieved by excluding groups of agents from the negotiation process, e.g., by dictating ontologies. Normally, this results in bad acceptance by these groups and a poor performance of the entire system.

- *Sharing Scope* vs. *Formality*: The relation between the sharing scope and the degree of formality is quite subtle. On one hand, being as explicit as possible when specifying a conceptualization is mandatory for sharing between several agents. Without a detailed understanding of a proposed ontology no commitments can be made by the agents, because probably there is no common interpretation of the proposal, and misunderstandings during application are predetermined. On the other hand, using a formal specification can be a hurdle for potential agents to participate in the sharing process. However, if formally less trained people are main addressees of an ontology, these users can not be excluded from negotiation and commitment. Often the acceptance of information systems fails, because the ontology is specified by highly trained designers in some formal language without taking into account that a less trained user may not comprehend the implications and therefore may not use the system in a proper way. For example, information representation in WWW search engines has a low degree of formality, but a large sharing scope, whilst powerful retrieval mechanisms are less common. In summary, being formal is a prerequisite to allow for sharing, but it inhibits a wide scope as it needs highly trained agents.

2.3. Techniques

Several modules can be used to balance a concrete information system within the design space spanned by the described dimensions:

- **Monitoring Services:** Analyzing the utilization of an ontology (e.g., feedback from a search machine) as well as monitoring the outer world can provide hints when the ontology should be re-engineered.

- **Responsibility Concepts:** In order to organize complex negotiation processes in large groups, it makes sense think about specific roles and responsibilities (thematic area managers, publishers, ...). These roles can be enacted by human as well as by machine agents [16].

- **Communication Support:** To achieve mutual understanding and generally agreed upon commitments, powerful discussion and negotiation services are required [6].

- **Formalization Services:** Informal–formal as well as tacit–explicit transitions are subject to different fields of research. Classical knowledge engineering provides human–centered methods and tools for both transitions [18]; contributions to informal–formal transitions on the basis of text documents come from information extraction, document analysis & understanding, and computational linguistics [17].

Comprehensive methodological and tool support for designing and maintaining enterprise KM ontologies throughout the whole lifecycle should comprise elements from all these areas. In the following we describe how ontology services — especially for domain ontologies — are provided in the FRODO framework.

3. Roles of Ontology–Related Actors

[19] propose a role–oriented analysis as a natural step in their Gaia methodology for agent–oriented design. In this section we perform such an analysis with respect to the ontology–related actors in an OM. In order to describe the various actors dealing with domain ontologies in FRODO we use the following dimensions:

- **Goals:** The actors operate in a regularly changing environment. In doing so, they not only react to such changes but also have their own goals and objectives which they try to achieve.

- **Knowledge:** Actors have knowledge with respect to the relevant realms of their environment, e.g., objects and other actors, as well as with respect to their own goals.

- **Competencies:** An actor's abilities to perceive and manipulate its environment and its own internal state. In a multi–actor environment, the abilities to communicate with other actors are particularly important.

Through communication, knowledge about facts, goals, competencies, etc. can be exchanged. This allows for negotiation and agreements which may lead to a distribution of tasks between actors, or to changes of an actor's knowledge and goals.

- **Rights:** Rights are a subset of an actor's competencies. They describe what an actor is allowed to do, e.g. read or manipulate an information item, or grant rights to other actors.

- **Obligations:** Obligations are also a subset of an actor's competencies. They describe what an actor is expected to do, e.g., due to a commitment in consequence of a complex negotiation procedure or because of an actor's intrinsic role.

The first three dimensions are the knowledge level descriptions proposed by Newell [14]. The latter two reflect that the various actors in distributed OMs form a society, not just an accumulation. Rights and obligations are the basis

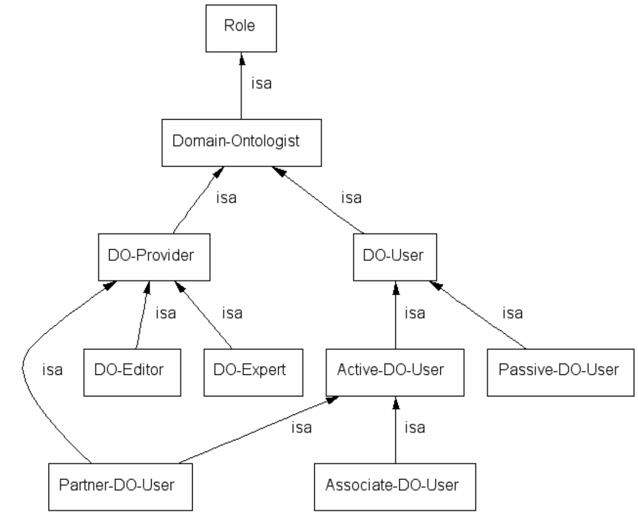

Figure 3. Taxonomie of Roles of Ontology–Related Actors.

for coordinating the negotiation processes that are needed to create a shared understanding.

Figure 3 shows a taxonomy of possible roles which ontology actors in a distributed organizational memory may take. The set of actors in an OM taking one of these roles with respect to a specific ontology form an ontology society. First, we distinguish between **ontology providers** and **consumers**. Ontology providers attend to the provision of ontology services (e.g. *experts* can answer queries about the relationship between two concepts) as well as to the acquisition and maintenance of a domain ontology (*editors*). Consumers, on the other hand, utilize a domain ontology in order to execute a specific application, e.g., find some knowledge items, annotate documents, etc.

These groups of actors typically have different goals with respect to an ontology. While consumers are only interested in completeness and soundness of an ontology with regard to their specific application, maintenance services take a more global view and claim these properties for the whole ontology.

Within the group of ontology consumers we distinguish between **active and passive users**. Passive users neither help to improve the ontology nor do they have any claims with respect to the ontology service. **Associates** also do not necessarily contribute to the ontology evolution, but have special quality requirements. Therefore, they are notified whenever the ontology changes. **Partners** commit to support the improvement of the domain ontology, hence they are both ontology consumers and providers. For the editor of an domain ontology, partners are of special importance as they are the main source for information about the utility of an ontology. However, the final responsibility for the ontology is in the **editor**'s hand.

Table 1 summarizes rights and obligations of the various user groups of a domain ontology regarding some typical ontology operations.

In the following we briefly sketch the competencies which are stated in Table 1:

- **Query:** All actors have the right to query an ontology service about properties of the domain. There may be different types of queries, e.g., about:
  - concepts: “Is a concept in the ontology?”, “Give a natural language description of a concept.”, ...
  - concepts and relationships: “Holds the relationship R between concept A and concept B?”, ...
  - ontologies: “Is ontology O1 a subontology of O2?”, ...
  - copy: “Give me a copy of ontology O and guarantee validity until revocation.”
- **Receive Update:** All actors but the passive ones have the right to be notified whenever a guarantee about the validity of an ontological information doesn’t hold any longer.
- **Suggest Update:** Clearly, any member of an “ontology community” can contribute to an improvement of the ontology. Partners and editors, in addition, commit to actively push ontology evolution.
- **Answer Queries:** To answer queries like the ones described above is one of the central tasks of an ontology service. The actor that attends this task is called *ontology expert*. An *editor* of an ontology is also able to answer these queries. However, he is not obliged to.
- **Edit:** Only editors can assert, modify and retract ontological propositions. As they have responsibility for the quality of an ontology, they are not forced to follow other actor’s suggestions. However, in order to obtain high acceptance and use of an ontology an editor will take all suggestions into consideration. Potentially, an editor has to coordinate a complex negotiation procedure between the actors to conceive his decision.
- **Send Update Notification:** An editor has the right and obligation to keep all given guarantees (e.g. with respect to an ontology’s validity) and notify the active users in case of changes.
- **Apply for Role:** This is a basic competence for joining an ontology society or changing an actor’s role within the society. The application is sent to an editor. This editor can then grant guarantees. Thereby the respective rights and obligations are negotiated.

	Non User	Passive User	Associate User	Partner User	Expert	Editor
Query		R	R	R	R	R
Receive Update			R	R	R	R
Suggest Update		R	R	R/O	R	R/O
Answer Queries					R/O	R
Edit						R
Send Update Notification						R/O
Apply for Role	R	R	R	R		
Grant Guarantees						R
Guarantee Quality						O

Table 1. Rights (R) and Obligations (O) of Ontology Actors.

- **Grant Guarantees:** e.g., validity for a certain time or until a certain event (cancellation), also the rights a user has when entering a ontology society.

- **Guarantee Quality:** Editors try to obtain a high quality of the domain ontology. Aspects of quality may be formal properties like soundness and completeness as well as "soft factors" like a good ratio between acquisition costs and use benefits. Guarantees about quality may be framed by a time interval or other constraints.

In summary, the previously described competencies can be grouped into three categories:

Ontology Utilization: Competencies like *Query* and *Answer Queries* are needed in the use phase of an ontology. Typical actors will be settled on the knowledge access level. A retrieval agent for example might exploit ontological knowledge to achieve higher recall and precision or to better present his results to the information consumer. Therefore it asks an ontology expert about the relation between two concepts.

Ontology Evolution: These competencies are necessary to negotiate ontology updates. E.g., if a retrieval agent takes the role of a partner user in an ontology society it might realize that information consumers often ask for information using a term that is not defined in the ontology. Hence, the retrieval agent would suggest the ontology editor to introduce a new concept. The ontology editor would thereupon coordinate a negotiation procedure between the active ontology users.

Ontology Socialization: Actors can join or leave an ontology society or they may change their role (e.g., from passive user to partner). In order to make a decision which role an actor wants to take it might need information about the content of an ontology and about the rights and obligations it has. Thus the affiliation in an ontology society might presuppose a complex negotiation procedure between the potential ontology user and the editor that grants guarantees.

As we deal with a multiple-OM scenario, we will have several domain ontologies. So, actors in this scenario can take one of the roles described above for each available ontology service. For example, the editor of domain ontology O1 might be an associate with respect to ontology O2 (e.g., the ontology in a different department). The *weakest role* "passive consumer" allows for a straightforward integration of external ontologies, because no severe commitments about rights and obligations are made.

4. Domain Ontologies in the FRODO Framework

In the introduction we argued for *vertical* scalability as well as for *horizontal* scalability of OMs. With respect to *domain ontology services* this requires facilities for both adding domain ontologies to an OM and accessing ontology services from other OMs.

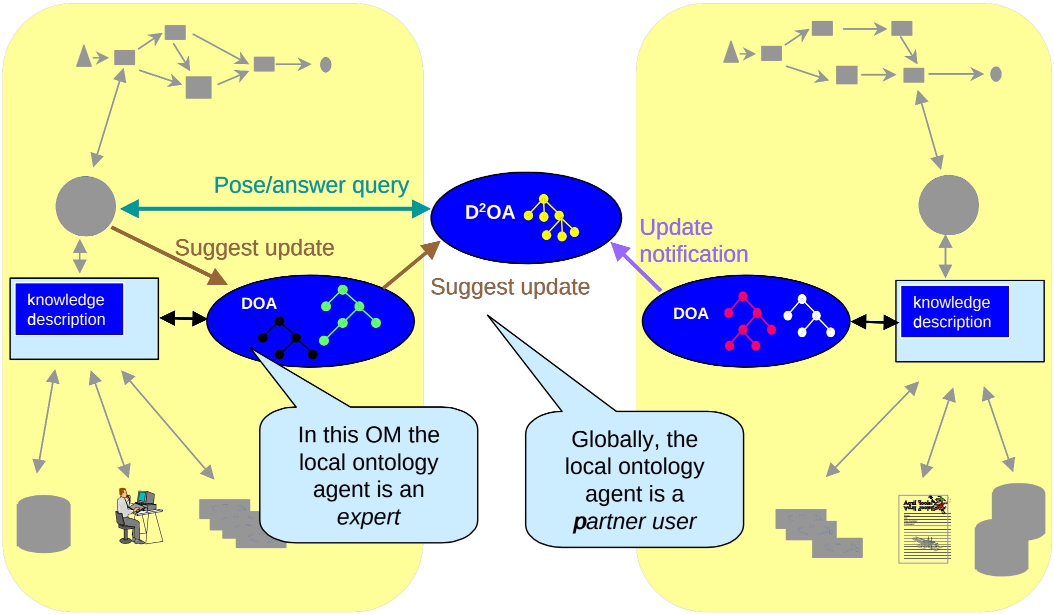

Figure 4. Sample Framework Instantiation with a D^{2}OA coordinating two OMs with local domain ontology agents (DOA).

Therefore we propose two types of ontology services: **Domain Ontology Agents** (DOA) and **Distributed Domain Ontology Agents (D^2OA).** Domain Ontology Agents are responsible for ontologies *within* one OM, Distributed Domain Ontology Agents are located *between several OMs* and facilitate cross–OM communication.

So, the task of D^2OAs is quite similar to “standard information integration ontologies” (e.g. mapping services), but much easier as the sources are already formal ontologies, not just “any information provider”.

Typical questions to DOAs are “What are the subconcepts of concept A?” whereas D^2OA answer questions like “Which OM contains concepts like A and B?” or “What does A mean in OM$_y$?”.

This structure better embraces the inherently distributed nature of (ontological) knowledge. Not *all conceptualizations* are shared between *all actors* of the system, but *ontology societies* are formed with respect to relevant domains. Additional infrastructure enables communication between these ontology societies.

Imagine for example two groups of experts, one for domain D1, one for domain D2. Each group negotiates its own domain ontology managed by DOA_{D1} and DOA_{D2}, respectively. D^2OA has knowledge what these ontologies are about and tries to identify points of contact or overlaps between them.

Then, D^2OA initiates a negotiation procedure between DOA_{D1} and DOA_{D2}. The result might be a common upper level ontology or a mapping for some parts of the ontologies.

DOAs as well as D^2OAs can be described in terms of the roles that have been outlined before. For their own ontologies they have the rights and obligations of *Ontology Experts* and *Ontology Editors*. DOAs are *Associate* or *Partner Users* of the D^2OA ontologies and vice versa.

In summary, the concept of ontology societies tries to find a *reasonable sharing scope* for portions of knowledge so that a common understanding is possible at all.

Figure 4 illustrates these ideas a bit: Here, we have two OM instances with their respective ontologies. In each of these OMs, there is an agent maintaining the local ontology, being an editor with respect to this local ontology. Information retrieval or information extraction agents within the two OMs may be partner users exploiting the ontological knowledge to perform their own services. They also may sometimes suggest ontology updates because they come too often to wrong answers or bad performance because of a mismatch between formalized ontologies and the evolution of the real world. If the local ontology agent decides to accept such an update suggestion and change the local ontology, all other agents actively using the ontology must be notified. Further, the global ontology agent should be notified in order to adapt mapping rules accordingly. It could also be the case that the local ontology agent, playing the role of a partner user with respect to the global ontology, might suggest to change the global ontology because specific local changes are so radical that this should be reflected in an update of the overall structures.

5. Outlook

In this paper we characterized possible roles of ontology-related actors in an OM scenario by specifying the respective rights and obligations in the ontology society. These role descriptions can be used as high–level specifications for the definition of agent types, speech acts, and standard services for an OM middleware.

In the FRODO project [2] we implement such a middleware on the basis of a FIPA-compliant agent platform [4]. An analysis of the FIPA specifications [7, 8] shows that such a platform provides a good infrastructure for the realization of the concepts presented above. We strive for further developments especially in the fields of knowledge representations and distributed inferencing services that take into account the specific requirements of an information landscape with various levels of reliability, depending on an actor's role [2].

The suitability of the FRODO approach is being tested in an application scenario in the realm of knowledge management for nuclear power engineering know–how. Here, knowledge is typically distributed over various sites (e.g., operators of power plants, public licensing authorities, several ministries), and a global view can not be obtained. Inevitably, a comprehensive use of knowledge is required to process critical procedures like the transport of nuclear material across the borders of states. An information infrastructure that allows each stakeholder of knowledge to keep his own view and sphere of responsibility on the one hand and defined zones of negotiated cooperation on the other hand can hopefully facilitate comprehensive knowledge management in such a delicate environment.

Acknowledgment This work was supported by the German Ministry for Education and Research, (Grant 01 IW 901, Project FRODO: A Framework for Distributed Organizational Memories) and the European Commission (Grant IST-1999-13002, Project DECOR: Delivery of Context–Sensitive Organizational Knowledge).

References

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, May/June 1998.

[2] A. Abecker, A. Bernardi, L. van Elst, A. Lauer, H. Maus, S. Schwarz, and M. Sintek. FRODO: A Framework for Distributed Organizational Memories. Milestone M1: Requirements Analysis and System Architecture. DFKI Document D-01-01, DFKI GmbH, March 2001.

[3] A. Abecker, S. Decker, and F. Maurer. Special Issue on Knowledge Management and Organizational Memory. *Int. Journal on Information Systems Frontiers (ISF)*, 2(3/4), 2000.

[4] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi agent systems with a FIPA-compliant agent framework. *Software – Practice And Experience*, 31:103–128, 2001.

[5] M. Bonifacio, P. Bouquet, and A. Manzardo. A distributed intelligence paradigm for knowledge management. In *Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes*. AAAI, Menlo Park, March 20–22, 2000.

[6] J. Domingue. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In *[10]*, 1998.

[7] FIPA. Abstract architecture specification. http://www.fipa.org/specs/fipa00001/.

[8] FIPA. Ontology service specification. http://www.fipa.org/specs/fipa00086/.

[9] M. Fox and M. Gruninger. Enterprise modelling. *AI Magazine*, Fall 1998.

[10] B. R. Gaines and M. A. Musen, editors. *11th Workshop on Knowledge Acquisition, Modeling and Management (KAW'98)*. Knowledge Science Institute, University of Calgary, Apr. 1998.

[11] T. R. Gruber. The role of common ontology in achieving sharable, reusable knowledge bases. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 601–602, San Mateo, CA, USA, April 1991. Morgan Kaufmann Publishers.

[12] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5,6):907–928, 1995.

[13] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36–56, 1991.

[14] A. Newell. The knowledge level. *Artificial Intelligence*, 18(1):87–127, 1982.

[15] D. O'Leary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, May/June 1998.

[16] F. Schmalhofer and L. van Elst. An oligo–agents system with shared responsibilities for knowledge management. In D. Fensel and R. Studer, editors, *Knowledge Acquisition, Modeling and Management, 11th European Workshop, EKAW '99*, pages 6379–384, Berlin, Heidelberg, New York, 1999. Springer.

[17] S. Staab, A. Maedche, C. Nedellec, and P. Wiemer-Hastings, editors. *ECAI'2000 Workshop on Ontology Learning*, Berlin, Germany, 2000. http://SunSITE.Informatik.RWTH-Aachen.de/-Publications/CEUR-WS/Vol-31/.

[18] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1–2):161–197, 1998.

[19] M. Wooldridge, N. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.