Chapter 1

Domain Ontology Agents in Distributed Organizational Memories

Ludger van Elst and Andreas Abecker
German Research Center for Artificial Intelligence (DFKI)
– Knowledge Management Department –
P.O. Box 2080, D-67608 Kaiserslautern, Germany
Phone: +49 631 205 3474, Fax: +49 631 205 32
eMail: {elst|aabecker}@dfki.uni-kl.de

Abstract: It is widely accepted that ontologies provide a useful means to facilitate access to, and reuse of knowledge in Organizational Memories. In *distributed OMs* – as the next evolution step for practical applications of OMs – the assumption of *globally shared* conceptualizations does not seem tenable. In order to retain the benefits of domain ontologies we propose to explicitly *control the sharing scope* of ontological knowledge. As an instrument to facilitate this we present the notion of *ontology societies* which are primarily defined by the rights and obligations of their members. We show an agent-based implementation of this concept and demonstrate the practical use of our approach in the FRODO architecture for Distributed Organizational Memories.

Key words: Ontologies, Distributed Organizational Memories

1. MOTIVATION

The vision of Knowledge Management (KM) encompasses the comprehensive use of an enterprise's knowledge, whoever acquired it, wherever it is stored and however in particular it is formulated. An Organizational Memory Information System, or Organizational Memory (OM) for short, is supposed to support this vision by accumulating, structuring, and fostering the utilization of explicit knowledge in manifold forms such as lessons learned entries, best practice documents etc. (Abecker et al., 1998). Technical support for Organizational Memories is often based on centralized approaches which seem well–suited to guarantee that *all* the information available is considered (Bonifacio et al., 2000). For instance, in the KnowMore framework (see Figure 3, (Abecker et al., 1998)), the problem of several heterogenous information sources is tackled by the introduction of a *uniform knowledge description level*: 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 (Abecker et al., 2000).

However, in real world scenarios centralized models often suffer from two drawbacks:

– *They neglect the advantages of the distributed nature of knowledge* (e.g. with respect to development and use) in an enterprise: It would be very expensive or even impossible to obtain a globally negotiated vocabulary many highly specialized actors. OMs could benefit from balancing both *local expertise* – which may result in non-globally shareable knowledge – and *overall views* on higher levels.
– *Centralized approaches are cumbersome in changing environments*: An OM's environment may for example change due to the reorganizations of an enterprise's structure. Furthermore, OM systems are typically not established all 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.

In the FRODO project[1], we propose the extension of the KnowMore framework towards a scalable, distributed OM technology (Abecker et al., 2001) which allows for evolutionary growth within one OM as well as for cooperation between several, independently introduced OMs. For such an OM technology the infrastructure to establish a shared understanding between the various actors is of crucial importance. The goal of this paper is to show the FRODO approach to handle domain ontologies in such a distributed OM scenario.

The remainder of this paper is organized as follows: In section 2 we give a brief overview of the dimensions of scalabality we envision in the FRODO framework and sketch the role of domain ontologies in such information landscapes. We argue that the handling of domain ontologies with respect to their sharing scope is especially important. Section 3 presents a role model of ontology-related actors in a distributed OM scenario which allows for explicit control of the sharing scope of ontological knowledge. In section 4 we show how these roles are implemented in FRODO and enacted to form

[1] http://www.dfki.uni-kl.de/frodo

ontological societies. The actor classes of the role model are candidates to become agent types and standard services in an envisioned agent-based middleware for the cost-effective off-the-shelf realization of Distributed Organizational Memories (DOMs). In section 5 the paper concludes with a brief sketch of an application example that will be used to show the practical suitability of our overall approach.

2. SCALABILITY ISSUES FOR DISTRIBUTED ORGANIZATIONAL MEMORIES

2.1 Vertical vs. Horizontal Scalability

In section 1 we motivated the notion of Distributed Organizational Memories (DOMs) by the distributed nature of knowledge in enterprises as well as with the pragmatic argument that OMs are typically introduced stepwise. A framework for DOM technology should support two types of scalability, namely *vertical scalability* that describes the ability of evolutionary growth *within one* OM and *horizontal scalability* as the cooperation *between several*, independently introduced OMs within the enterprise.

Assuming a KnowMore-like OM architecture (cf. Figure 1), vertical scalability means

– integration of new services in order to meet additional requirements on the application level (e.g. complementing retrieval services by a task-oriented summarization service),
– incorporation of additional legacy systems in the source layer and
– extension / change of underlying ontologies for the knowledge description layer.

For the purpose of this paper, we concentrate on horizontal scalability. For example, separate OMs may exist for different departments of an enterprise (design, production, customer relationship management, etc.). In order to "globally optimize" the whole product lifecycle, information has to be used across the departments. Therefore, a communication means between several OMs is 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.

The following simple example that is based on a rather generic OM architecture motivates several modes of horizontal OM integration and shows a general tradeoff between complexity of communication and quality of service.

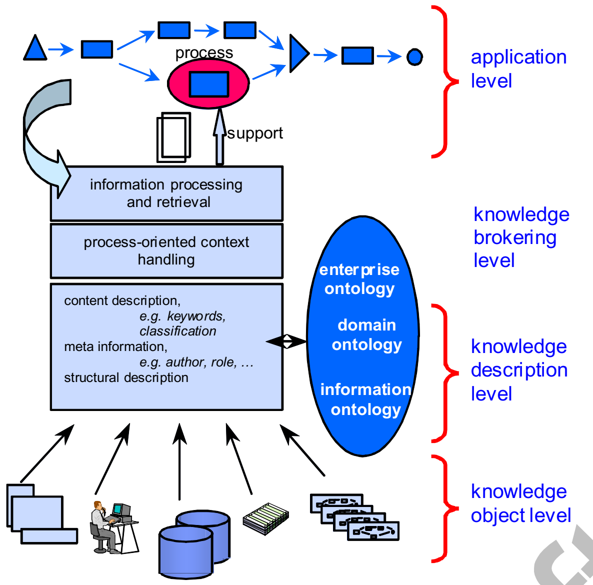

Figure 1. A Reference Architecture for a Single OM (derived from (Abecker et al., 1998))

2.2 Horizontal Integration: A Simple Example

Imagine two "simple OMs". OM_1 consists of

1. a set of documents, e.g. books *(knowledge object level)*
2. that are arranged on bookshelves with category labels *(knowledge description level),*
3. an intelligent information assistant who can select relevant books with respect to a query *(knowledge brokering level),* and
4. a researcher working in a project *(application level).*

OM_2 is a library and is organized quite similarly: Documents are arranged on bookshelves according to the "ACM classification". There are assistants to the "chief librarian", helping to organize the library, select new books, write recommendations for literature etc.

The standard process within OM_1 is described as follows: The researcher working on a special work package of the project has a specific information need (e.g. about agent–oriented software architectures, AOS). He therefore asks his assistant to fetch the relevant documents for this topic from the shelf. As the assistant is intelligent he not only fetches the documents on the shelf labelled “AOS”, but also documents from other shelves labelled with related topics (e.g. “FIPA specifications”). Furthermore, he sorts these documents according to specific criteria (e.g. software platforms, text books etc.), briefly summarizes them, and leaves documents in the shelf that are definitely outdated. The result is presented to the researcher.

Perhaps the documents on his own bookshelf do not satisfy the researcher's information need on AOS. Hence, he asks his assistant to fetch more information from the library (OM_2). Now the information assistant (OM_1) can cooperate with the library (OM_2) on various levels:

– **Object Level:** The assistant knows about a specific book and just fetches it from the library.
– **Knowledge Description Level:** The assistant searches for documents on AOS. He uses the catalog of keywords to find out which categories of the ACM classification are appropriate, and then fetches the books.
– **Knowledge Brokering Level:** He asks one of the library assistants to suggest and fetch him a set of relevant books. In order to get a good result, he has to explain the term “AOS” to the library assistant and tell him other criteria for “relevant books”. Then the library assistant performs an extensive search for literature, compiles the documents and delivers them to the researcher's information assistant.

Furthermore, cooperation between the library and the researchers can happen on even higher levels:

– **Application Level:** The workplan for the researcher's project includes a task “ordering of project literature”. Therefore, he has to cooperate with the “chief librarian” who manages a global budget for literature.
– **Second Order Processes:** In order to establish a really useful libary and not to buy all the relevant literature locally in the projects, every six month the library commission – consisting of some researchers and the chief librarian – meets and defines a strategy for the purchase of books and magazines.

While the lower-level cooperations only require very little communication effort, the higher levels rely on communication between the various agents (e.g. “What are the goals for a department library?”, “What does the term AOS mean?”). On the other hand, the latter services typically are of better quality and higher impact: Suggestions by a good technical librarian are more precise and relevant than books found by a pure keyword search or the set of all books labelled with a particular ACM classification code.

This simple example points out that – in order to select a specific form of cooperation between OMs – a *tradeoff* has to be balanced between *complexity of communication* and the *quality of service*. While the direct exchange of information objects on a "syntactical basis" – e.g. referenced by its unique identifier – is quite cheap in terms of communication effort, the lack of semantic information holds down the quality of service; e.g. it is not clear whether the object really provides the information desired. On the other hand, high quality services at the knowledge brokering level typically need more communication, e.g. to negotiate the intended meaning of a request, its costs etc.

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. In knowledge management it is often desirable to have this shared understanding explicitly represented and it is widely accepted that ontologies provide an adequate means for this purpose (cf. Fensel, 2001).

2.3 Ontologies in Distributed OMs

Ontologies (Gruber, 1991) are commonly employed in OMs to facilitate (human or machine) access to, and reuse of knowledge in OMs. Ontologies

- provide views and navigation structures for manual browsing (O'Leary, 1998, McGuiness, 1998);
- facilitate natural language access (Guarino *et al.*, 1999);
- provide background knowledge for query expansion or query rewriting (Bodner and Song, 1996, Sintek *et al.*, 2000);
- enable management of non-textual media (Khan and McLeod, 2000); and
- support retrieval and integration of information from different, distributed sources (Staab *et al.*, 2000a, Heflin and Hendler, 2000).

In virtually all these scenarios, ontologies are the basis for articulation of information demands by information consumers, or for characterization of information offers by information providers. Furthermore, all of them are multi–actor scenarios by nature, and they make great demands with respect to flexibility, extensibility, and maintainability in a changing world. Keeping in mind that an OM is an enterprise information system which has to constantly survey its cost-benefit assessment, the question arises what the basic design decisions are for constructing such systems, whether there are trade-offs influencing the cost-benefit ratio, and how to stabilize certain operating points. In (van Elst & Abecker, to appear), we discuss basic dimensions of information in an OM: stability, degree of formalization, and sharing scope. We describe the rather subtle interactions between these dimensions. Some of these interactions are depicted in figure 2.

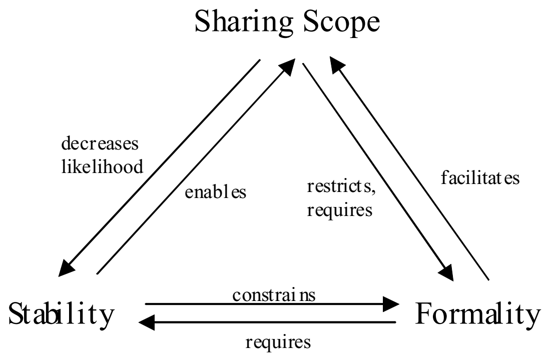

Figure 2. Tradeoffs between sharing scope, stability, and formality of information in KM systems (cf. (van Elst & Abecker, 2001) for a detailed discussion)

From this analysis we argue that comprehensive methodological and tool support for designing and maintaining ontologies throughout the whole lifecycle should comprise not only formalization services (Studer et al., 1998, Staab et al., 2000b) but also stability monitoring and explicit control of the sharing scope. In the remainder of this paper we focus on the latter. The notion of *ontological societies* we elaborate in the next section is proposed as an instrument to facilitate control of sharing scope through commitment to specific rights and obligations with respect to an ontology.

3. ROLES FOR ONTOLOGY–RELATED ACTORS

Responsibility concepts are central to our approach to distributed OMs: In order to organize complex negotiation processes in large groups, it makes sense to think about specific roles and responsibilities (thematic area managers, publishers, ...). These roles can be enacted by human as well as by machine agents (Schmalhofer and van Elst, 1999).

3.1 Characterization of Actor Roles

(Wooldridge et al., 2000) propose 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. In this section we perform such an analysis with respect to ontology–related actors. In order to describe the various actors dealing with domain ontologies 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 similar to the knowledge level descriptions proposed by (Newell, 1982). The latter two reflect that the various actors form a society, not just an accumulation. Rights and obligations are the basis for coordinating the negotiation processes that are needed to create a shared understanding.

3.2 A Taxonomy of Roles

Figure 3 shows a taxonomy of possible roles which actors in an ontology-based information system may take. The set of actors taking one of these roles with respect to a specific ontology forms 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.

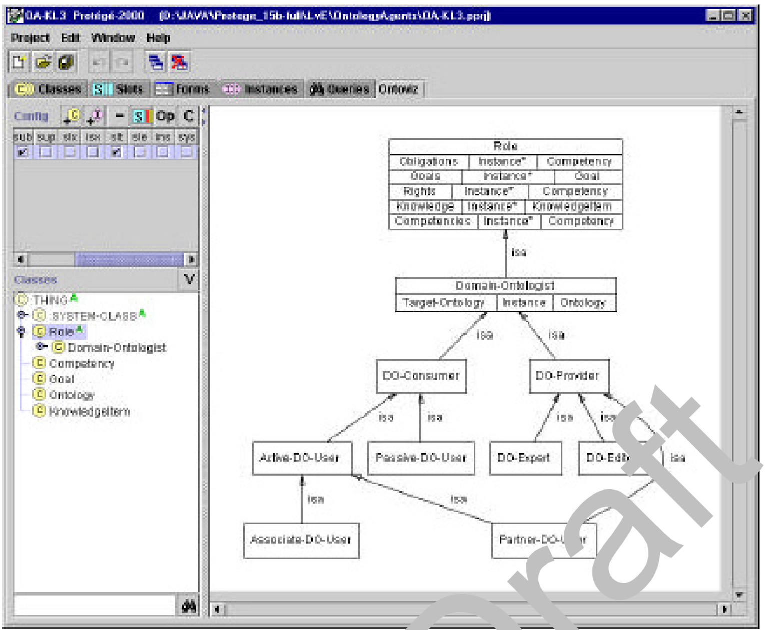

Figure 3. Taxonomy of Roles of Ontology-Related Actors

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 of information about the utility of an ontology. However, the final responsibility for the ontology is in the editor's hand.

3.3 Competencies of Actors

In the following, we describe typical competencies of ontology-related actors taking one of the previously described roles with their respective rights and obligations. These competencies concern ontology utilization, evolution, and the forming of ontological societies.

3.3.1 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.

– **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
  a) *concepts*: “Is a concept in the ontology?”, “Give a natural language description of a concept.”, ...
  b) *concepts and relationships*: “Does the relationship R hold between concept A and concept B?”, ...
  c) *ontologies*: “Is ontology O1 a subontology of O2?”, ...
  d) *copy*: “Give me a copy of ontology O and guarantee validity until revocation.”
– **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.

3.3.2 Ontology Evolution

Competencies of this category 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 to the ontology editor the introduction a new concept. The ontology editor would thereupon coordinate a negotiation procedure between the active ontology users (cf. Bailin & Truszkowski, 2001).

- **Receive Update:** All actors but the passive ones have the right to be notified whenever a guarantee on 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.
- **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.
- **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.

3.3.3 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 negotiation procedure between the potential ontology user and the editor that grants guarantees.

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

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

	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.

4. IMPLEMENTATION OF THE ROLE MODEL IN AN AGENT-BASED FRAMEWORK

In this section we show how the concepts previously presented are realized within the FRODO framework (Abecker *et al.*, 2001). As a technical basis of FRODO we use the JADE agent platform (Bellifmine *et al.*, 2001) that provides us with the mechnism to handle distributeness and allows to easily plug in new services. Extensions were made to utilize Protégé (Noy *et al.*, 2000) as the core knowledge representation for agents, especially for ontology representation. This means the platform can have multiple instances of Protégé wrapped as agents.

4.1 Competencies as Speech Acts

In section 3.3 various competencies of actors for ontology utilization, evolution and socialization were described. FRODO defines speech acts that implement these competencies similarly to FIPA: The sender, receiver and content of a speech act are specified; feasibility preconditions contain the qualifications oft the act; the rational effect shows the reasons for which an act might be selected; FIPA speech acts are used to define the semantics of the FRODO act[2].

[2] The definition by FIPA speech acts should not necessarlily been seen as operationalization of a FRODO speech act. In fact this is done in order to ground FRODO by the semantic model FIPA provides.

FRODO speech act	ApplyForRole
Description	An agents wants to take a specific role in a society and therefore sends am application to the editor.
Sender	S
Receiver	R
Content	role, society
Feasibility Precondition	NOT(Believes(S, hasRole(S, society, role))) AND Wants(S, role, society)
Rational Effect	Believes(R, Wants(S, role, society)))

FIPA_action	`(inform`
	`:sender S`
	`:receiver R`
	`:content Wants(S, role, society))`

Comment	Alternative specification:
	`(request-when`
	`:sender S`
	`:receiver R`
	`:content (action (R, GrantRole(S, role, society))`
	`(Believes(R,possibleRole(S, society, role)))`

FRODO speech act	GrantRole
Description	The editor of a society gives an applicant a specific role, i.e. commits to the respective rights and obligations.
Sender	E
Receiver	AP
Content	role, society
Feasibility Precondition	Believes(E, Wants(AP, role, society) ANDhasRole(E, society, Editor) AND Believes(E, possibleRole(AP, society, role))
Rational Effect	Believes(E, hasRole(AP, society, role)), Believes(AP, hasRole(AP, society, role))

FIPA_action	`(inform`
	`: sender E`
	`: receiver AP`
	`: content Believes(E, hasRole(AP, society, role))),`
	`(inform`
	`: sender E`
	`: receiver E`
	`: content Believes(E, hasRole(AP, society, role)))`

Comment	The second inform just ensures the Rational Effect „Believes(E, hasRole(AP, society, role))“.

Table 2. Two Examples of FRODO Speech Acts for Ontology Socialization.

Table 2 shows two examples of FRODO speech acts for forming ontology societies. With *ApplyForRole* an agent expresses the intention to take a specific role in a society. In the table two alternative specifications are given: a) In the simple specification the sender just wants the receiver to know that it wants to take the role and therefore the semantics of *inform* is used. Here, the receiver itself must infer that an appropriate reaction might be a *GrantRole* or a *Deny*. b) The second alternative is much more specific. Here, a *request* for a *GrantRole* action is used. This action should be applicable as soon as the receiver believes the desired role is possible for the sender. The precondition for *ApplyForRole* is that the sender really wants that role in the respective society and that it not already believes to have the role.

Accordingly, the precondition for a *GrantRole* is that the sender

- has the right to do so (hasRole(sender, society, Editor)),
- has a belief that the receiver wants the role, and
- the specific role is appropriate for the receiver.

So the editor of an ontology is repsonsible for forming the ontology society by granting roles to other agents. The operationalization of a role's rights and obligations for a concrete agent is done by a social layer in FRODOs agent platform. The technical details go beyond the scope of this paper.

4.2 Domain Ontology Agents in FRODO

In order to achieve both vertical and horizontal scalability of OMs, we need facilities for both adding domain ontologies to an OM and accessing ontology services from other OMs. We propose two types of ontology-related software agents for the distributed OM implementation:

1. **Domain Ontology Agents** (DOA) are responsible for ontologies *within* one OM
2. **Distributed Domain Ontology Agents** (D^2OA) 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 maybe a bit 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^2OAs 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[3]. 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.

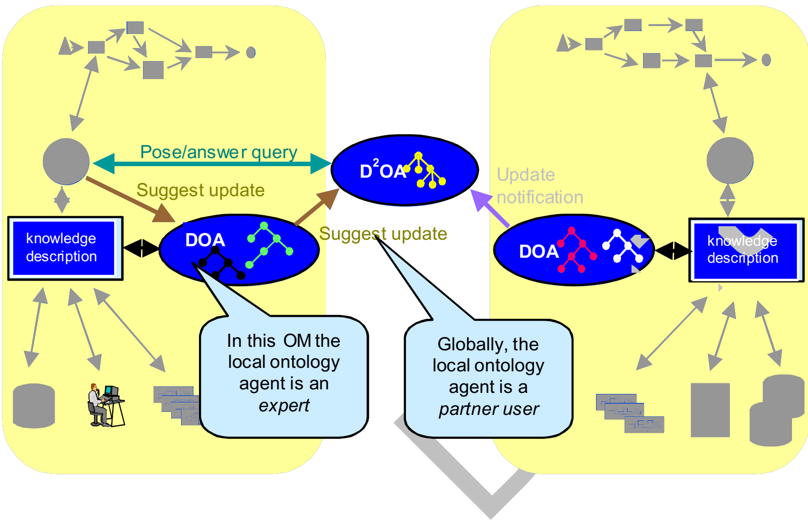

Figure 4. D^2OA Coordinating Two OMs With Local Domain Ontology Agents (DOA)

[3] Obviously D^2OA has to cope with all the well-known and hard problems of ontology mapping. This topic is beyond the scope of this paper. Actually, in FRODO we have hand-coded mapping rules and try an instance/text categorization-based method to detect overlaps.

Figure 4 illustrates these ideas: 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 too often come 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 must be notified which actively use the ontology. 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 changing the global ontology because specific local changes are so radical that this should be reflected in an update of the overall structures.

5. SUMMARY AND OUTLOOK

In this paper we motivated the need for Distributed Organizational Memories (DOM) which in comparison to centralized approaches

– better reflect the distributed nature of knowledge in a company,
– are more adaptable to changing company environments, and
– can be introduced successively.

We discussed the basic modes of scalability in a DOM and identified a tradeoff between complexity of communication and quality of service in the cooperation of several OMs (section 2). Here, domain ontologies seem to form an important operating point. The role model for ontology-related actors in a DOM (section 3) is motivated by the fact that an ontology (and the information described by it as well) in an information system can normally not be seen as an eternal truth. It is rather a socially constructed artefact which is used as a tool with a given purpose, which obeys the laws of economic rationality, and which evolves over time. In the FRODO project, we used the role descriptions with their rights and obligations as high–level specifications for the definition of agent types, speech acts, and standard services for a DOM middleware (section 4). We implemented such a middleware on the basis of a FIPA-compliant agent platform (Bellifemine *et al.*, 2001).

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 cannot 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

Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M. (1998). Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3):40–48, May/June 1998.

Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M. (2000). Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project. *Int. Journal on Information Systems Frontiers*, 2(3/4):139–162, 2000.

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

Bailin, S.C., and Truszkowski, W. (2001). Ontology negotiation using JESS. In *Proc. 3rd Int. Conference on Enterprise Information Systems, ICEIS-2001, Setúbal, Portugal*, 2001.

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

Bodner, R.C., and Song, F. (1996). Knowledge-based approaches to query expansion in information retrieval. *Canadian Conference on AI*, pp. 146–158, 1996.

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

van Elst, L., and Abecker, A. (2001) Ontology-Related Services in Agent-Based Distributed Information Infrastructures. In: *Proceedings of the Thirteenth International Conference on Software Engineering & Knowledge Engineering*, June 13-15, 2001, Buenos Aires, Argentina, pp. 79-85.

van Elst, L., and Abecker, A. (to appear) Ontologies for Information Management: Balancing Formality, Stability, and Sharing Scope. *Expert Systems with Applications*, to appear.

Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, 2001.

Gruber, Th.R. (1991). 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.

Guarino, N., Masolo, C., and Vetere, G. (1999). Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, May/June 1999.

Heflin, J., and Hendler, J. (2000). Searching the web with SHOE. In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop*. AAAI Press, 2000.

Khan, L., and McLeod, D. (2000). Effective retrieval of audio information from an-notated text using ontologies. In S.J. Simoff and O.R. Zaiane, editors, *Workshop on Multimedia Data Mining (MDM/KDD'2000), August 2000, Boston, MA.*, 2000.

McGuiness, D.L. (1998). Ontological issues for knowledge-enhanced search. N. Guarino (Ed.). *Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy*. IOS Press, 1998.

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

Noy, N. F., Fergerson, R. W., and Musen, M. A. (2000). The knowledge model of Protégé-2000: Combining interoperability and flexibility. *12th International Conference on Knowledge Engineering and Knowledge Management* (EKAW'2000), Juan-les-Pins, France, 2000.

O'Leary, D. (1998). Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3):34–39, May/June 1998.

Schmalhofer, F., and van Elst, L. (1999). 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.

Sintek, M., Tschaitschian, B., Abecker, A., Bernardi, A., and Müller, H.-J. (2000). Using ontologies for advanced information access. In J. Domingue, editor, *PAKeM 2000, The Third International Conference and Exhibition on The Practical Application of Knowledge Management, Manchester, UK*, 2000.

Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R., and Sure, Y. (2000a). AI for the web - ontology-based community web portals. In *AAAI/IAAI*, pp. 1034–1039, 2000.

Staab, S., Maedche, A., Nedellec, C., Wiemer-Hastings, and P. (Eds). (2000b). *ECAI'2000 Workshop on Ontology Learning*, Berlin, Germany, 2000.

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

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