Domain Ontology Agents in Distributed Organizational Memories

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 3470, Fax: +49 631 205 3210
e-Mail (elst,aabecker)@dfki.de

Abstract

Ontologies are used in agent–oriented software development, information systems, and expert systems in order to support interoperability, declarativity, and intelligent services. In the FRODO project, we design a scalable, agent–based middleware for distributed Organizational Memories (OM). In this paper, we investigate which ontology–related services should be provided as middleware components. To this end, we discuss three basic dimensions to characterize stored information that determine the concrete specification of ontology–based systems, *formality*, *stability*, and *sharing scope*. A short discussion of techniques which are suited to find a balance on these dimensions leads to a characterization of roles of ontology–related actors in the OM scenario which are described 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 Motivation

In knowledge management (KM), it is widely accepted that ontologies as explicit specifications of conceptualizations [Gruber, 1991; 1995; Neches *et al.*, 1991] 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 [O'Leary, 1998]. In the context of Organizational Memory (OM), ontologies provide a vocabulary for specifying information resources as well as information needs [Abecker *et al.*, 2000a; 2000b] in order to evolve from a keyword–based towards a concept–based information management, indexing, and retrieval approach. They also form the basis for knowledge–enhanced or knowledge–assisted search and retrieval [Sintek *et al.*, 2000; McGuinness, 1998].

In these applications, ontologies serve as formally represented "specifications of discourse in the form of a shared vocabulary" [O'Leary, 1998]. Such a shared understanding is particularly important because KM typically deals with multi–actor scenarios (cf. [Perez *et al.*, ; Silverman and Owens, 1996; Gandon *et al.*, 2000]). 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. Technical support for such a vision is often based on *centralized approaches* which seem well–suited to guarantee that the *complete* information available is considered [Bonifacio *et al.*, 2000]. For instance, in OM reference architecture (Figure 1) derived from the KnowMore framework [Abecker *et al.*, 1998], the problem of several heterogeneous 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.

In the FRODO project [Abecker *et al.*, 2001] we aim at extending the centralized KnowMore framework towards a distributed OM scenario. This is motivated by the drawbacks of centralized models:

1. *They neglect the advantages of the distributed nature of knowledge (e.g. with respect to development and use) in 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.

2. *Centralized approaches are cumbersome in changing environments*: An OM's environment may for example change due to reorganizations of an enterprise's structure. Furthermore, OM systems are typically not established at once for a whole company, but introduced step by step 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: *vertical scalability* describes the stepwise extension of *one* OM as shown in Figure 1 by additional elements on each level, like new business processes at the application level, new information sources at the knowledge object level, etc. *Horizontal scalability* means that *different* OM systems should
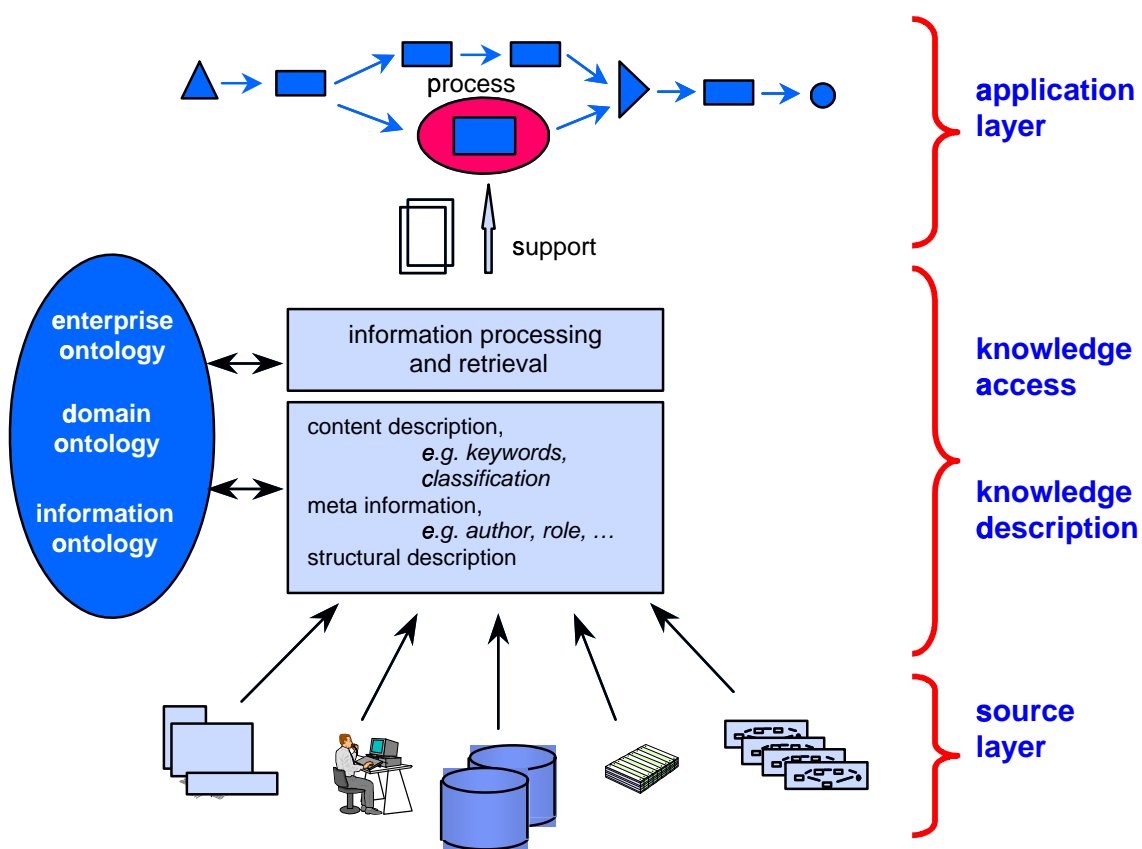
Figure 1: OM Framework (derived from the KnowMore architecture)

be interoperable for cooperative query answering to a high degree.

In this paper we describe our considerations about design issues and basic decisions for OM software architectures which are able to cope with both types of scalability. In the next section, we present in some more detail the idea of scalable OMs. In section 3, coming from the definition of an ontology, we encourage a comprehensive view on three dimensions of information that have fundamental impact on the usefulness of ontology–based KM systems: formality, stability, and sharing scope. We point out the trade–offs involved when determining a concrete operating point in the respective spectrums, and list some techniques/research areas dealing with scaling along the respective dimensions. These considerations lead to at least two conclusions: (i) it makes sense to think about scalable OM systems as agent–based information systems with many scaling services realized as agent services; and (ii) domain ontologies in such a complex scenario should not only be created, maintained and evolved by a single ontology agent, but rather should be managed by the coordinated collaboration of a number of OM actors dealing with specific issues. In section 4, we elaborate this idea top down: starting with the knowledge level, we characterize roles of ontology–related actors with respect to their knowledge, goals, competencies, rights, and obligations and sketch an ontology of ontology–related actors. This is the basis for

a list of ontology–related services in an ontology society — which later will be provided by specific speech acts between ontology–related agents in the OM middleware. These services can be grouped in ontology evolution, ontology utilization, and ontology socialization services. A small example of the intended software realization of these concepts illustrates the FRODO approach. In section 5, we conclude with a short outlook on our next steps.

2 Scalability Issues for Distributed Organizational Memories

In this section, we describe a rather general framework for scalability of distributed OMs. A stylized "real world scenario" elaborates on the various cooperation potentialities in the intended information landscapes.

1. Vertical scalability describes the ability of evolutionary growth within one OM:
   - integration of new services in order to meet additional requirements
   - incorporation of additional legacy systems
   - extension/change of underlying ontologies

2. Horizontal scalability means the cooperation between independently introduced OMs within one enterprise. For example, there may exist separate OMs for different departments of an enterprise (design, production,

costumer relation, etc.). In order to "globally optimize" a complete 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: for instance, cooperative information gathering [Nodine *et al.*, 1999] or intelligent information integration [Wiederhold, 1996] are examples for a horizontal OM integration on the knowledge brokering level. Cross–organizational workflows are an example 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. In the following, we illustrate the horizontal integration aspect with a small example.

Horizontal OM Integration: Complexity of Communication vs. Quality of Service

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.

The following simple example illustrates this tradeoff. Imagine two "simple OMs":

OM_1 consists of

1. a set of documents, e.g. books (*information 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 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 with respect to 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 with respect 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 in his own bookshelf do not satisfy the researcher's information need about 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 library 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.

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. This is where ontologies show their particular importance. Coming from the definition of ontologies, three dimensions for characterizing information in an OM can be identified. These are presented in more detail in the next section.

3 Three Dimensions for Characterizing Information in an OM

Information in OM information systems can be described with respect to various dimensions, e.g., granularity, trust-

worthiness, 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 outsources 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 enter-prise 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 [Studer *et al.*, 1998] 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 KM system, a detailed understanding of their interactions is useful.

Interactions between the Three Dimensions

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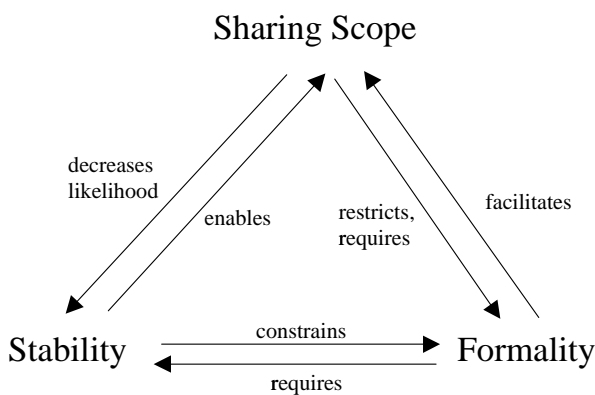

Figure 2: Sharing scope, stability, and formality of information in KM systems

- *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 [Abecker *et al.*, 1998], 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 slight, 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 [Fox and Gruninger, 1998]). 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.

Assessment of 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 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].

- *Communication Support*: To achieve mutual understanding and generally agreed upon commitments, powerful discussion and negotiation services are required [Domingue, 1998]. Here, research on computer supported cooperative work offers valuable contributions (cf. [Borghoff and Schlichter, 2000; Krauss and Fussell, 1990]).

- *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 [Studer *et al.*, 1998]; contributions to informal–formal transitions on the basis of text documents come from information extraction, document analysis & understanding, and computational linguistics [Staab *et al.*, 2000]. From a cognitive point of view it seems promising to support various degrees of formality for intelligent information provision (cf. [van Elst and Schmalhofer, 1999; Schmalhofer and van Elst, 2000]).

While the first (monitoring services) and the last (formalization services) item refers to the stability and the formality dimension of information, respectively, the middle two concepts (responsibilities and communication) address the sharing aspect of ontologies. They will lead later on to our approach of an ontology society of ontology–related agents. 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, with specific emphasis on the several agents' rights and responsibilities which are the basis for forming a society from a set of individuals.

4 Roles of Ontology-Related Actors

In this section, we identify classes of ontology–related actors in a distributed OM by describing their roles. Wooldridge et al. propose such a role-oriented analysis as a natural step in their Gaia methodology for agent–oriented design [Wooldridge *et al.*, 2000], especially when it is manifest to take an organizational view on the application scenario.

In order to characterize the various actors dealing with domain ontologies in FRODO we use the following criteria:

- *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 has 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 criteria correspond to the knowledge level descriptions proposed by Newell [Newell, 1982]. The latter two reflect that the various actors in distributed OMs 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.

Figure 3 shows a taxonomy of possible roles which ontology actors in a distributed organizational memory may take.
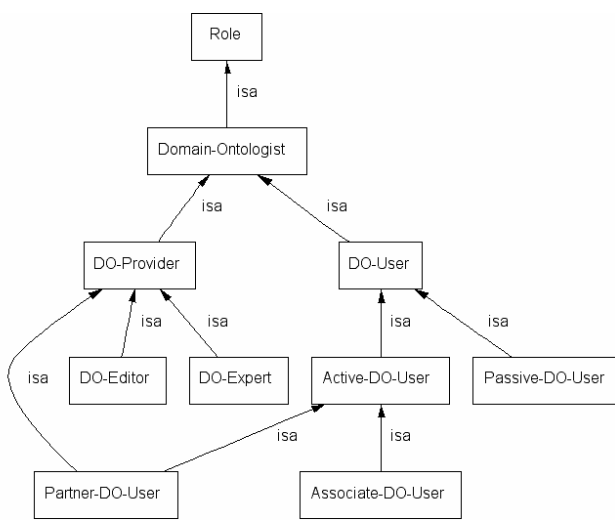
Figure 3: Taxonomie of Roles of Ontology–Related Actors.

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 a 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.

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 A equivalent to ontology B?", "Is ontology A a sub-ontology of B?", ...
    – copy: "Give me a copy of ontology A 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 doesnt 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 responsibilty 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 guarantee (e.g. with respect to an ontology's validity) and notify the active users in the 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.

- 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 constrains.

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 brokering level. A retrieval agents for example might exploit ontological knowledge to achieve higher recall

	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.

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 there upon 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 A might be an associate with respect to ontology B (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. The following sketch of a sample software framework may illustrate the basic architectural approach a bit.

Domain Ontologies in the FRODO Framework

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

Therefore we propose two types of ontology services: Domain Ontology Agents (DOA) and Distributed Domain Ontology Agents (D²OA). 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 D²OAs 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 D²OAs 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. D²OA has knowledge what these ontologies are about and tries to identify points of contact or overlaps between them.

Then, D²OA 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.
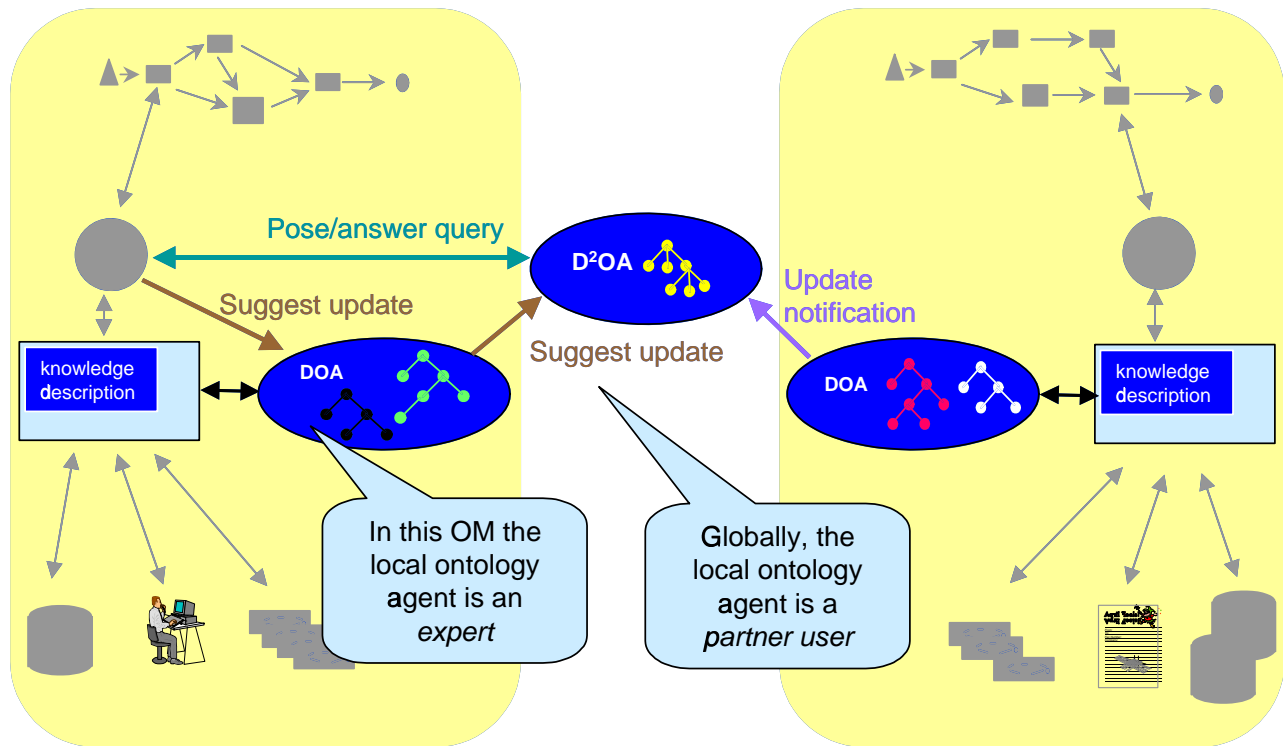
Figure 4: Sample Framework Instantiation with a D^{2}OA coordinating two OMs with local domain ontology agents (DOA).

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

DOAs as well as D^{2}OAs 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^{2}OA ontologies and vice versa (see figure 4) .

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 [Abecker *et al.*, 2001] we implement such a middleware on the basis of a FIPA–compliant agent platform [Bellifemine *et al.*, 2001]. An analysis of the FIPA specifications [FIPA, a; b] 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 [Abecker *et al.*, 2001].

The suitability of the FRODO approach is being tested in an application scenario in the realm of knowledge manage-

ment for nuclear power engineering knowhow. 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.

References

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

[Abecker *et al.*, 2000a] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Context–aware, proactive

delivery of task–specific knowledge: The knowmore project. *Int. Journal on Information Systems Frontiers (ISF)*, 2(3/4):253–276, 2000.

[Abecker *et al.*, 2000b] Andreas Abecker, Ansgar Bernardi, Heiko Maus, Michael Sintek, and Claudia Wenzel. Information supply for business processes: Coupling workflow with document analysis and information retrieval. *"Knowledge-Based Systems" Journal, Special Issue on Artificial Intelligence in Knowledge Management, Elsevier*, 13, 2000.

[Abecker *et al.*, 2001] 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.

[Bellifemine *et al.*, 2001] 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.

[Bonifacio *et al.*, 2000] 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.

[Borghoff and Schlichter, 2000] U. M. Borghoff and J. H. Schlichter. *Computer–Supported Cooperative Work*. Springer, Berlin, Heidelberg, New York, 2000.

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

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

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

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

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

[Gandon *et al.*, 2000] F. Gandon, R. Dieng, O. Corby, and A. Giboin. A multi–agent system to support exploiting an xml–based corporate memory. In *[Reimer, 2000]*, 2000.

[Gruber, 1991] Th. 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.

[Gruber, 1995] Thomas 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.

[Krauss and Fussell, 1990] R. M. Krauss and S. R. Fussell. Mutual knowledge and communicative effectiveness. In R. E. Kraut J. Galegher and C. Egido, editors, *Intellectual Teamwork: Social Foundations of Cooperative Work*, pages 111–146. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[McGuinness, 1998] D. L. McGuinness. Ontological issues for knowledge-enhanced search. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 302–316. IOS Press, Washington, DC, 1998.

[Neches *et al.*, 1991] Robert Neches, Richard Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36–56, 1991.

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

[Nodine *et al.*, 1999] M. H. Nodine, J. Fowler, T. Ksiezyk, B. Perry, M. Taylor, and A. Unruh. Active information gathering in infosleuth. In *Symposium on Cooperative Database Systems for Advanced Applications*, pages 15–26, 1999.

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

[Perez *et al.*,] Comma Consortium, Ph. Perez, H. Karp, R. Dieng, O. Corby, A. Giboin, F. Gandon, J. Quinqueton, A. Poggi, and G. Rimassi, et al. Corporate memory management through agents. http://citeseer.nj.nec.com/381229.html.

[Reimer, 2000] U. Reimer, editor. *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management, PAKM2000*, Basel, Switzerland, 2000. http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-34/.

[Schmalhofer and van Elst, 1999] 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.

[Schmalhofer and van Elst, 2000] F. Schmalhofer and L. van Elst. Cognitive analyses within increasingly larger organizational contexts. In J.M. Schraagen, S.F. Chipman, and V.L. Shalin, editors, *Cognitive Task Analysis*, pages 303–316, Mahwah, New Jersey, London, 2000. Lawrence Erlbaum Associates.

[Silverman and Owens, 1996] B. G. Silverman and J. Owens. Enterprise information infrastructure (eii): Design issues and guidelines. *Information and Systems Engineering Journal*, 2(1):19–45, 1996.

[Sintek *et al.*, 2000] Michael Sintek, Bidjan Tschaitschian, Andreas Abecker, Ansgar Bernardi, and Heinz-Jürgen Müller. Using ontologies for advanced information access. In John Domingue, editor, *PAKeM 2000*, April 2000.

[Staab *et al.*, 2000] 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/.

[Studer *et al.*, 1998] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1–2):161–197, 1998.

[van Elst and Schmalhofer, 1999] L. van Elst and F. Schmalhofer. A cooperative comprehension assistant for intranet-based information environments. In M. Klusch, O. M. Shehory, and G. Weiss, editors, *Cooperative Information Agents III, Proceedings of the Third International Workshop, CIA'99*, pages 390–401, Berlin, Heidelberg, New York, 1999. Springer.

[Wiederhold, 1996] G. Wiederhold, editor. *Intelligent Integration of Information*. Kluwer Academic Publishers, Boston, 1996.

[Wooldridge *et al.*, 2000] M. Wooldridge, N.R. 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.