# Agent-based IT support for Knowledge Management

Manuel Kolp

IAG - Information Systems Research Unit - University of Louvain, 1 Place des Doyens, B-1348 Louvain-La-Neuve, Belgium, tel.: 32-10 47 83 95, kolp@isys.ucl.ac.be

**Abstract.** Knowledge Management (KM) involves information technology (IT) aspects as well as psychological and business management issues. A KM effort can always only be successful, if a holistic perspective is taken. By taking a holistic perspective on KM IT support, we identify a number of core services and components of a KM IT support system. The identified components are organized into an IT framework. In this paper, we propose software agent technology as the key paradigm for that framework proposal. One central service in that proposal, the ontology service, is considered in detail. Among other issues, creation, application and representation of ontologies in the framework are explored and areas for future research are identified.

## 1 Introduction

There is a lot of hype around *Knowledge Management* (KM), although KM is not something new. Simply put, KM is a collection of methodologies and strategies for business process re-engineering to foster the flow of knowledge between people in an organization. Organizations have always struggled to manage the knowledge of their employees as effectively as possible. Groupware support systems and workflow systems and before that, management information systems have contributed a great deal to managing knowledge in the past. However, what has created the hype around KM are the altered preconditions.

- Technology has been improved greatly, such that information management is no longer the pressing issue it used to be. Information has become a commodity, but the need for applicable knowledge has increased drastically. The focus is now more on how people interact with systems.

- The possibilities of world-wide information management make world-wide communities of interest and practice possible. This creates an enormous potential for synergies, the key for which is the generation of community knowledge and thus again, social issues in the community.

- The large number of information sources available creates the need for methodologies to create relevant, important and applicable knowledge from available information for members of an organization. The lack of common context among sources of information makes this an important issue.

- A faster-paced business world with quickly-changing administrative frameworks (virtual organizations) brings with it the need for effective conservation of knowledge.

The need for effective KM in organizations has been largely recognized. A great number of KM efforts have been under way in industry. Most of the KM efforts have been technology focused. Moreover, a plethora of proprietary and incompatible software applications have been used. 80-90% of these narrow, technology-focused efforts failed [24]. There are two main reasons for the failure. On the one hand, pedagogical and psychological aspects of knowledge creation have neither been considered enough in the design of the applications nor in the actual integration into the respective business processes. On the other hand, existing proprietary information management tools block the exchange of information between applications and users. This prevents what has been termed the *knowledge flow* and identified as the key leverage to KM in [3]. The knowledge flow describes the collection, distribution, recombination of knowledge and creation of new knowledge.

One *lesson learned* from these efforts is, that a holistic approach is required in the design of IT infrastructures as well as the actual business process re-engineering for a successful KM effort. The *second lesson learned* is, that some components of a KM IT support system have to be centralized, easily accessible and thus allow services to exploit synergies among applications.

Knowledge is considered as something that resembles information functionally applicable by the person who holds the respective knowledge. Consequently, knowledge can only exist in a human being. Thus, IT can support KM by managing information in a way that effectively supports the core processes involved in KM efforts. The framework proposed here supports knowledge development and knowledge distribution. However, due to the flexible nature of the framework proposal, other processes can be integrated.

Different classes of knowledge have been identified. A central notion in KM is the difference between *explicit knowledge* and *tacit knowledge*, first introduced in [17,18]. Explicit knowledge can be easily explicitly coded (e.g. as natural language) into information and can be characterized as theoretical knowledge. Tacit knowledge is hard to express and can be characterized as experience. In [16], the transfer process between the different kinds of knowledge is analyzed. In [3] this is referred to as the *knowledge life-cycle*. Both [3] and [16] extend the analysis of Anderson's *ACT-theory* [22], which is limited to the transfer of explicit knowledge to tacit knowledge. With a strict definition of tacit knowledge, only explicit knowledge can be stored in an IT system. With a less strict definition of tacit knowledge, there is no principal reason why tacit knowledge should not be explicable. It is more an economic question, since explication of tacit knowledge involves considerable effort . In any case, an IT system can also offer support for the transfer of tacit knowledge. In our framework we will aim at supporting the exchange of explicit as well as tacit knowledge. We will not present any support for knowledge explication.

Existing groupware systems support team knowledge management preferably for administratively static environments. With floating administrative frames (virtual organizations) and a growing number of relevant information sources outside the administrative scope of an organization, knowledge management support has to become more flexible, open and loosely-coupled. Existing knowledge management support systems feature the ability to access a great number of information sources. However, they lack openness in communication among components and ease of integration of new components. The rapidly growing Internet information market will sooner or later rule out systems, into which new components cannot easily be integrated.

We propose a flexible and high-level IT support for KM. Our approach leads us to propose an agent-based framework that includes vital components of an open KM architecture. We will derive essential services for KM IT support and present our agent-based software architecture in Section 2. Different aspects of the ontology support service will be discussed in Section 3.

## 2 Essential KM support services

We will present a number of what we consider essential services for KM support. We will first characterize three characteristic groupings of people in and among organizations and derive the support services for these groupings of people. For each grouping, different classes of knowledge will be considered. We consider the individual, the team and the community as the important groupings for KM. We use those terms according to what has been presented in [21]. The characteristics we focus on are:

- Team: small grouping of people, members know each other, common goal, shared artifacts, administrative fixed rules
- Community: members do not necessarily know each other, members have something in common (spatial relations, interest, social conventions), sense of membership and boundary [13], ongoing rhythm of social interaction [15].

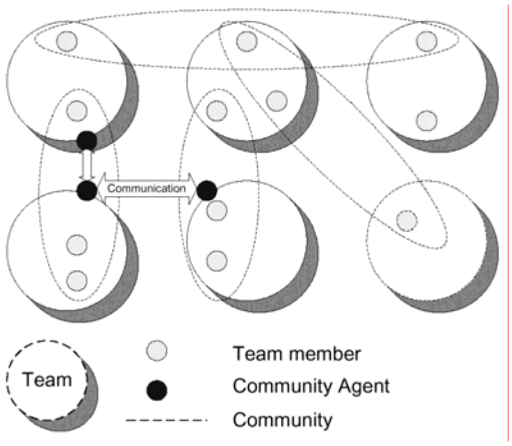


**Fig.1**. ; Relation of user, team and community

An individual member of an organization can be a member of several teams in the organization as well as several intra-organizational and inter-organizational communities. In teams, a knowledge flow without restrictions is desired, whereas in communities, restrictions might apply. The relation of user, team and community is shown in Figure 1.

### 2.1 Why Agents ?

The services in our framework are offered by agencies, which include several agents each. Each agent offers several services. The agents are clustered into the *user agency*, the *team agency* and the *community agency*. Within each agency, there are several agents that offer certain services. We use the software agent paradigm [23] for our architecture for several reasons:

- the software agent paradigm is a powerful modeling paradigm for distributed systems that allows to model autonomous components and mobility of components.

- high-level *Agent Communication Languages* (ACL) (FIPA ACL, KQML) allow for a loose coupling of components and easy integration of new components.

- communication over ACL reduces the total number of interfaces between components and is open for integration of new applications.

Figure 2 shows our architecture. The agencies and the services offered by the agents in the architecture will now be motivated.

## 2.2 Services for the individual

All services for the individual are grouped in the user agency in our framework (Figure 2).

*User information management services*
To allow the user to manage his personal information, a KM support system has to support structured storage and retrieval of personal information. Examples for such support are file systems or databases that can be keyword searched or browsed. Ideally, the services will support reuse of information modules [20]. The user information management services reacts on demand pull by the user. The user information management services are provided by the *user item agent* in the *user agency*.

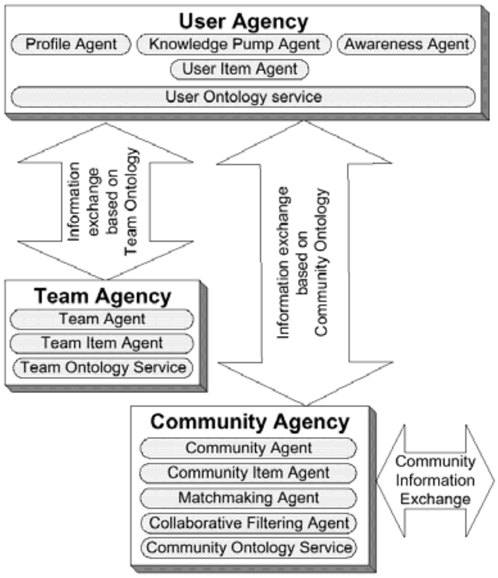


**Fig. 2** : Architecture of a KM IT system

*Pro-active push of information* In order to perform a task, knowledge is required. The more relevant knowledge is available, the better that task can be performed. The planning phase of a task can be considered as the decisive phase, when existing knowledge is taken into account and used for the solution of a task. Thus, the more relevant knowledge is available in the planning phase, the more potential problems can be identified and potential mistakes avoided. The respective information to create relevant new knowledge has to be made available *pro-actively*. The reason why the system should *autonomously push* information to the user lies in human information demand behavior (Figure 3).

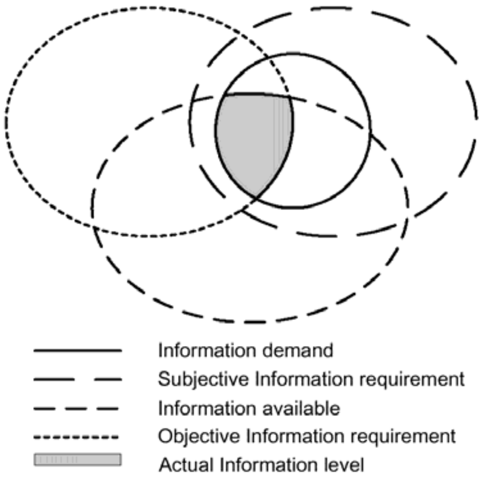


**Fig. 3** : Human information demand behavioral model

The *subjective information requirement* is what the user considers the information relevant to perform the task. *The objective information requirement* is what is really relevant information to perform the task. The *information available* is the information the user can access at the moment. The *information behavior* of the user leads to the *user's information demand*, which is only a subset of what the subjective requirement is. The intersection of the four sets of information is the *actual level of information* the user has available to create new knowledge for the performance of the task. We identify two ways to increase the information level with the user. One way is to make the user more aware of the information available, and thus enlarge the subjective information requirement. The other way is to supply information that has not been demanded, but is assumed to be of relevance. We do not think however, that the third way, determining the objective information requirement for a task, is feasible for an IT system. There are several other advantages to pro-active information provision. For example double work can be avoided and new knowledge is continuously created. In order to provide information pro-actively, an IT system has to act partly *autonomous*. We term the autonomous component in our framework (Figure 2) the *knowledge pump agent* [3]. In order to find and query the varying sources of information in communities, the knowledge pump agent accesses *information brokerage services* in the team and community agents. Available information sources in the teams and communities are registered with the brokerage services. However, it is clearly not desirable to push all available information to the user. Thus, pushed information should be restricted to information that is relevant for the user's task.

*Provision of relevant information*

To increase the actual level of relevant information with the user, an IT system has to reason about the user's current task and the required relevant information as far as possible. Principally, there are two ways to extract the user's current task: by observation of all interaction of the user with the system or by explicit provision of the work context by the user. Ideally, in order to reduce the cognitive load on the user induced by the system to a minimum, all information is automatically collected. However, in practice, the information collected will not suffice for higher-level semantic reasoning about the user's task. Thus, a combination of explicit provision and automatic collection of task information should be targeted. In our framework, the task relevance information is managed by the *awareness agent*. In order to push relevant information to the user, the knowledge pump agent queries the awareness agent for task information and then queries the broker services for relevant information.

*Personalization of provided information* Another important aspect for knowledge acquisition is the accordance of the personal context and the context of provided information. The better knowledge is personalized to the user's context the better it is acknowledged and internalized by the user [11]. Information provided to the user should be *personalized to the user's assumed qualification and interest*. A negative effect that can not be avoided, is that if new knowledge contrasts an established belief of the user, it is most probably not internalized. In order to achieve personalization, *a user profile* has to be kept. The profile models interests and qualification of the user. Moreover, media preferences for information presentation can be stored in the user profile. In our framework, the *user profile agent* manages the user profile. The knowledge pump agent queries the awareness agent for the current work context, then the profile agent for qualification and preferences of the user. With this information, the user can be provided with tailored, task-relevant information presented in a way that suits the user's preferences. The user profile can hold information explicitly given by the user and information retrieved from the awareness agent, for example on documents that the user read or published. More detailed recommendations on how user profiles can be designed can be found in [14] and [9].

*Information retrieval services*
All the above services rely on information retrieval services. We consider information retrieval of any kind as a transfer of information between separate organizational memories (OM) [1]. The consequence is loss of context. This loss of context is not a problem when personal info of the user is retrieved, since the user's context stays the same. Generally, there are two ways of context adoption: 1) The information is either stored together with some reference context that is common to a grouping and can later be used to access the relevant information. 2) The context is reconstructed as far as possible from the stored information. Since support services for groupings are not yet considered, context has to be extracted from the information available and mapped to the user context. This task can be accomplished by text mining techniques or query-by-example techniques. However, the reconstructed context is never as rich as the original context. Information retrieval services are included in the knowledge pump agent in our framework. The knowledge pump agent uses the brokerage services in the team agent and the community agent to find respective information sources.

### 2.3 Services for a team

The team services are not described as broadly as the user services and the community services.

*Shared workspace services*
For a team, a shared workspace has to be provided. The workspace should allow sharing of documents and other information items. The document management services must support geographically dispersed teams and manage versioning and consistency of the shared information space. The shared workspace service functions as an information source for the individual services. The items in the shared workspace are annotated with meta-information about the common team context. With this meta-information, information can be more easily managed and retrieved. The document management services in our framework are included in the *team item agent* in *the team agency*. The team item agent also offers the possibility to publish certain information items to communities or keep them private to the team. The team agency also includes brokerage services for the information sources registered with the team agent.

*Awareness services*

To actively promote knowledge sharing among team members, information about the other team member's current work has to be provided. This is how double work can be avoided, a context for the individual's work can be given and shared knowledge creation can be fostered. In [5] awareness information is sub-classified into

1. *workspace awareness*: information about the shared workspace.
2. *group-structural awareness*: infor-mation about the structure of the team.
3. *social awareness*: information about emotional status and competencies of team members.
4. *informal awareness*: information about the environment of a team member.

Since team members know each other, group-structural awareness as well as social awareness is existent in team members. However, workspace awareness and informal awareness can be of great use, especially for geographically dispersed teams. For example, if one team member discovers an important document and stores it in the shared workspace this should be made aware to all team members. Awareness events are collected in the user agency and partly published to the *team agency*. From there awareness information to all team members is available as status information and events. Examples for awareness services are NESSIE [19] and Elvin [10].

*Shared Ontology services*
For an effective shared workspace service, a common classification scheme for documents is required. The classification scheme gives a formal and explicit definition of concepts in the domain of knowledge of a team. With respect to this common classification scheme, documents can be stored and easily retrieved. The set of common concepts can be characterized as a kind of common vocabulary and is often referred to as *ontology* [12]. In a team, a shared ontology makes it easier to organize information. However, due to the close relation of team members, an explicit specification is only in some cases required, since with time, the team will develop a common understanding and vocabulary anyway. All team members profit from the shared ontology in terms of information retrieval, since all information that has been mapped to the team context already can then be easily accessed by other team members.

### 2.4 Services for a community

Communities have really only become possible through the global Internet. In communities with large numbers of members there is enormous potential for knowledge sharing synergy.

*Community information management services*
In order to exploit synergies in a community to a maximum, the community members have to be registered with the *community agent*. The community agent also offers brokerage services for available information sources in the community. Additionally, it is important to store information of interest for all community members with the *community item agent*. The information stored with the community item agent can for example be ratings by users for information items, which are later exploited for collaborative filtering services.

*Matchmaking services*
Community members do not know each other. Thus, one important community support service is to find a community member with certain characteristics. The basis for matchmaking is the personal user profile of each user or shared annotated items in the community. Matchmaking services can facilitate the flow of knowledge in two ways: 1) Through direct expert mediation for the exchange of tacit knowledge. In this case, explicitly given profile information is matched against community members and an expert in a certain field can be found. 2) Through indirect exploitation of matching

user profiles. The user's own profile is matched against other users' profiles. In the case of a good match, information items that each of the users considers valuable can be exchanged with the matched user. The profile information matched can be information in the user profile or item rating information from the community item agent. The matchmaking services in our framework are provided by the *matchmaking agent* in the *community agency*.

*Collaborative filtering services* Collaborative filtering relies on matchmaking services and a repository of information rated by a user. Upon a query for information, profile data of two users or community item rating data is matched. Information that is not rated by both of the matched users is delivered to the respective other user. The quality of information retrieval can be improved, if only positively rated information is exchanged. If collaborative filtering information is *pushed* to the user, the process is called collaborative recommendation.

*Awareness services*
Awareness services for communities have to provide the user with information about the boundaries and the existence of the community. This refers to membership information as well as information about shared items in the community. Awareness services are provided by the *community agent* in our framework.

*Shared ontology services*
The shared ontology service is especially important for a community. An ontology is essentially the basis for the user profiles in a community and if the community members do not share an ontology, profile-based services will not be available to the users. Moreover, if community members agree on an ontology for information classification, most loss of context during information transfer between organizational memories can be avoided. A shared ontology also provides a powerful basis for community awareness services. Management of information is facilitated with the classification scheme of an ontology.

## 3 Ontology service

In the last section we introduced our framework and identified the shared ontology service as a central and very important service. We will now consider this service in some more detail. We first explain with an example what an ontology is. We will then present concepts for the application, creation and representation of ontologies and introduce some existing ontology projects. An ontology is an explicit specification of a conceptualization of a knowledge domain. Ontologies consist of concepts and relations among the concepts. Examples for concepts are elephant or animal. An example relation between those two concepts would be the isa relation: elephant isa animal. Concepts are characterized by attributes and so are implicitly the relations among concepts. For example if the concept elephant is characterized by the attributes grey and all attributes that the concept animal has, the isa relation between the two is implicitly defined. Thus, an ontology provides a taxonomy aswell as relations among the words in the taxonomy.

### 3.1 Ontology applications

In [2], ontologies are used for information classification and retrieval. In our framework, ontologies are required as the basis for the user profile, for information management services, community awareness services and agent communication.

*User profiles*
Using an ontology, the qualification or interests of a user can be approximated by marking concepts in the ontology. Single concepts or subgraphs of the concept graph can be marked as known or of

interest for a user to represent the profile. Based on the concept structure of an ontology, a query for an expert in a certain field can be posed. If there are no common and commonly rated items in a community, this is also the basis for collaborative filtering and recommender services.

*Information classification*
With the concept space of an ontology, information items in a community or a team can be classified according to the ontology. This provides for a reduction of the loss of context that is inevitable with the transfer between organizational memories [1]. A lot of the original context can be stored as meta-information with the information item. Later, a query can be posed according to the ontology structure to retrieve information that is relevant in terms of user interest, task context and also qualification.

*Visualization*
In a community, the interests and qualifications of members can be visualized according to the ontology scheme. This contributes to the community awareness. Information spaces can be presented graphically according to the ontology scheme. This also makes it easy for the user to browse in information spaces.

*Agent communication*
Another important application of ontologies is agent communication. This application is most important in communities. In teams, there is often the possibility to enforce a proprietary mode of communication among involved software components of a KM support system. However, in communities, this is not possible. In order to reduce complexity of syntactic compatibility of communication, agent communication languages such as *FIPA ACL* or *KQML* have been created. This ensures a mini- mum of compatible communication among components of a system. However, for semantic coherence of communication, a shared ontology is required. The possibility to successively enhance ontologies allows the components to adopt their communication abilities to the other components.

## 3.2 Ontology creation

Ontologies can be created manually or automatically from a base of information. Manual creation can be done by an expert in the field or collaboratively by members of a team or a community. Collaborative creation increases acceptance among the members and decreases maintenance effort. Automatic creation requires a set of documents from which concepts can be extracted. This can be done with existing text mining techniques or other semantic clustering techniques.

### 3.3 Ontology representation

Ontologies can be represented in different languages. One example is KIF [7]. In [6] ontologies are represented based on frame logic. As a serialized way of ontology representation, an XML application language could be proposed.

### 3.4 Related ontology projects

The Ontolingua system [8] is a powerful tools to design ontologies. The On2broker project is a comprehensive ontology project [6]. In [4] a semantic net is built on the basis of a description logic to conceptualize objects in the domain of tourism. This is an approach that could be easily transferred for an ontology implementation. In [12], a number of design principles for ontologies are introduced. We consider the shared ontology and the user agency the parts of our architecture, where most research still has to be done.

## 4 Conclusion

We found that existing KM efforts either suffer from patchwork IT solutions or from non-acceptance of IT solutions that were not designed with a holistic perspective on KM. We derived core services of KM IT support by taking a holistic perspective on KM and related research. We found that user, team and community are the important grouping concepts for KM and that most potential for KM lies in community support as well as user support (computer human interaction). The identified support services were transferred into an IT system framework. Agent technology was identified as the key system design paradigm. Among the services, we identified ontology services as one of the most important services. Aspects of ontology applications, creation and representation were discussed. Areas for future research on ontologies were identified.

## References


1. M. Ackerman and C. Halverson. Reexamining organizational memory. *Communications of the ACM*, 43(1):pp. 59-64, 2000.
2. V. Benjamins, D. Fensel, and A. Pérez. Knowledge management through ontologies. In *Proc. of the 2nd Int. Conf. on Practical Aspects of Knowledge Management (PAKM98)*, Basel, Switzerland, 1998.
3. U. Borghoff and R. Pareschi, editors. *Information Technology for Knowledge Management*. Springer Verlag, 1998.
4. J. Bullock and C. Goble. TourisT: The application of a description logic based semantic hypermedia system for tourism. In *Proc. of the 9th ACM Conf. on Hypertext and hypermedia*, pp. 132-141, Pittsburg, USA, 1998.
5. P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proc. of the ACM Conf. on Comp. Supported Cooperative Work*, pp. 107-114, New York, USA, 1992.
6. D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, and A. Witt. On2broker: Semantic-based access to information sources at the WWW. In *Proc. Of the World Conference on the WWW and Internet (WebNet 99)*, Hawai, USA, 1999.
7. M. Genesereth and R. Fikes. KIF: Knowledge Interchange Format version 3.0 reference manual. Technical report, Stanford University, Knowledge Systems Laboratory, 1992.
8. R. Fikes, A. Farquhar, and J. Rice. Tools for assembling modular ontologies in ontolingua. Technical report, Stanford University, Knowledge Systems Laboratory, 1997.
9. J. Fink, A. Kobsa, and A. Nill. User-oriented adaptivity and adaptability in the Avanti project. In *Proc of the Conf. Designing for the Web: Empirical Studies*, Microsoft Campus, Redmond, USA, 1996.
10. G. Fitzpatrick, T. Mansfield, S. Kaplan, D. Arnold, T. Phelps, and B. Segal. Instrumenting and augmenting the workaday world with a generic notification service called Elvin. *In Proc. of the 6th European Conference on Computer Supported Cooperative Work (ECSCW99)*, pp. 431-451, Copenhagen, Danemark, 1999.
11. J. Forgas. *Interpersonal behaviour*. Pergamon Press, New York, USA, 1985.
12. T. Gruber. *Toward principles for the design of ontologies used for knowledge sharing*. Technical report, Stanford University, Knowledge Systems Laboratory, 1993.
13. T. Ishida. *Community Computing*. John Wiley and Sons, 1998.

14. A. Kobsa and W. Wahlster, editors. *User Models in Dialog Systems*. Springer Verlag, London, 1989.
15. E. Mynatt, A. Adler, M. Ito, and V. Oday. Design for network communities. *In Proc. Of the ACM Conf. on Human Factors in Computing Systems (CHI97)*, pp. 210-217, Atlanta, USA, 1997.
16. I. Nonaka and H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, Oxford, UK, 1995.
17. M. Polanyi. *Personal Knowledge*: *Towards a Post Critical Epistemology* Chicago, USA, 1958.
18. M. Polanyi. *The Tacit Dimension.* Routledge & Keel Paul, 1966.
19. W. Prinz. NESSIE: An awareness environment for cooperative settings. In *Proc. of the 6th European Conference on Computer Supported Cooperative Work (ECSCW99)*, pp. 391-411, Copenhagen, Danemark, 1999.
20. C. Süÿ, B. Freitag, and P. Brössler. Meta-modeling for Web-Based teachware. In *International Workshop on the World-Wide Web and Conceptual Modeling (WWWCM 99)*. Paris, France, 1999.
21. J. Schlichter, M. Koch, and C. Xu. Awareness - the common link between groupware and community support systems. In T. Ishida, editor, *Community Computing and Support Systems*, pages pp. 77-93. Springer Verlag, 1998.
22. M. Singley and J. Anderson. *The transfer of cognitive skills*. Cambridge Press, 1989.
23. G. Weiss, editor. *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.
24. R. Whiting. Myths & realities – What is behind one of the most-misunderstood IT strategies. In *Information Week online*, November 1999,
25. http://www.informationweek.com.