Chapter XIX

Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

Aurora Vizcaino
University of Castilla-La Mancha, Spain

Juan Pablo Soto
University of Castilla-La Mancha, Spain

Javier Portillo
University of Castilla-La Mancha, Spain

Mario Piattini
University of Castilla-La Mancha, Spain

ABSTRACT

Efforts to develop Knowledge Management have increased in recent years. However, many of the systems implanted in companies are still not greatly used by the employees because the knowledge that these systems have is often not valuable or on other occasions, is useful but employees do not know how to search for that which is most suitable. Moreover, employees often receive too many answers when they consult this kind of systems and they need to waste time evaluating all of them in order to find that which is most suitable for their necessities. On the other hand, many technical aspects should also be considered when developing a multi-agent system such as what knowledge representation or retrieval technique is going to be used. To find a balance between both aspects is important if we want to develop a successful system. However, developers often focus on technical aspects giving less importance to knowledge issues. In order to avoid this, we have developed a model to help computer science engi-
neers to develop these kinds of systems. In our proposal, first we define a knowledge life cycle model that, according to literature and our experience, ponders all the stages that a knowledge management system should give support to. Later, we describe the technology (software agents) that we recommend to support the activities of each stage. The chapter explains why we consider that software agents are suitable for this end and how they can work in order to reach their goals. Furthermore, a prototype that uses these agents is also described.

INTRODUCTION

In the last decades, knowledge management has captured enterprises’ attention as one of the most promising ways to reach success in this information era (Malone 2002). A shorter lifecycle of products, globalization, and strategic alliances between companies demand a deeper and more systematic organizational knowledge management. Consequently, one way to assess an organization’s performance is to determine how well it manages its critical knowledge.

In order to assist organizations to manage their knowledge, systems have been designed. These are called Knowledge Management Systems (KMS), defined by Alavi and Leidner (2001), as IT-based systems developed to support/enhance the processes of knowledge creation, storage/retrieval, transfer, and application.

However, developing KMS is a difficult task; since knowledge per se is intensively domain dependent whereas KMS often are context specific applications. Thus, reusability is a complex issue. On the other hand, the lack of sophisticated methodologies or theories for the extraction of reusable knowledge and reusable knowledge patterns has proven to be extremely costly, time consuming and error prone (Gkotsis, Evangelou et al. 2006). Moreover, there are several approaches towards KMS developing. For instance, the process/task based approach focuses on the use of knowledge by participants in a project or the infrastructure/generic system based approach focuses on building a base system to capture and distribute knowledge for use throughout the organization (Jennex 2005). On the other hand, before developing this kind of system it is advisable to study and understand how the transfer of knowledge is carried out by people in real life. However, when developing KMS developers often focus on the technology without taking into account the fundamental knowledge problems that KMS are likely to support (Hahn and Subramani 2000).

Different techniques have been used to implement KMS. One of them, which is proving to be quite useful, is that of intelligent agents (van Elst, Dignum et al. 2003). Software agent technology can monitor and coordinate events or meetings and disseminate information (Wooldridge and Jennings 1995). Furthermore, agents are proactive in the sense that they can take the initiative and achieve their own goals. The autonomous behavior of the agents is critical to the goal of this research since it can reduce the amount of work that employees have to perform when using a KM system. Another important issue is that agents can learn from their own experience. Consequently, agent systems are expected to become more efficient with time since the agents learn from their previous mistakes and successes (Maes 1994).

Because of these advantages different agent-based architectures have been proposed to support activities related to KM (Gandon 2000). Some architectures have even been designed to help in the development of KMS. However, most of them focus on a particular domain and can only be used under specific circumstances. What is more, they do not take into account the cycles of knowledge in order to use knowledge management in the system itself. For these reasons, in this paper we propose a generic model for developing KMS. Therefore, in section two we describe the model.
and the software agents that we propose to support it. In section three, we explain how the agents are structured and how they have been modeled using the INGENIAS methodology (Pavón and Gómez-Sanz 2003). Later, section four describes an scenario to illustrate how agents collaborate to reach a common goal. Section five summarizes related works carried out with agents. Finally, conclusions and future work are outlined in section six.

Table 1. Knowledge Life Cycle

<table>
<thead>
<tr>
<th>Model</th>
<th>Stage1</th>
<th>Stage2</th>
<th>Stage3</th>
<th>Stage4</th>
<th>Stage5</th>
<th>Stage6</th>
<th>Stage7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonaka and Takeuchi (1995)</td>
<td>Socialization</td>
<td>Externalization</td>
<td>Combination</td>
<td>Internalization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wiig (1997)</td>
<td>Creation</td>
<td>Storing/ gathering</td>
<td>Use</td>
<td>Leverage</td>
<td>Sharring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Davenport and Prusak (1998)</td>
<td>Generation</td>
<td>Codify/ Coordinate</td>
<td>Transfer</td>
<td>Roles and Skills</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tiwana (2000)</td>
<td>Acquire</td>
<td>Sharing</td>
<td>Use</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alavi and Leidner (2001)</td>
<td>Creation</td>
<td>Storage/ Retrieval</td>
<td>Transfer</td>
<td>Application</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rus and Lindvall (2002)</td>
<td>Creation/ Acquisition</td>
<td>Organization/ Storage</td>
<td>Distribution</td>
<td>Application</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nissen (2002)</td>
<td>Creation</td>
<td>Organization</td>
<td>Formalize</td>
<td>Distribute</td>
<td>Application</td>
<td>Evolve</td>
<td></td>
</tr>
</tbody>
</table>
indicates what process a KMS should support (see Figure 1). This is a focus different to the previous one based on describing the knowledge cycle in human being and/or in companies.

The stages of our proposal are: acquisition, storage, use, transfer and evaluation. The first three stages are considered in most knowledge life cycles (see Table 1). We have added transfer (also considered in several cycles) and evolution. The former because a KMS should disseminate knowledge to those people that can need it. The latter because knowledge should always be updated otherwise it would not be used.

In the following paragraphs each stage of the model is described. At the same time and with the goal of illustrating that it is possible to support each stage by using current technology, we are going to explain how a software agent could be implemented for a KMS.

a. **Knowledge acquisition** is a key component of a KMS architecture. This stage includes the elicitation, collection, and analysis of knowledge (Rhem 2006). During this process, it is vital to determine where in the organization the knowledge exists and how to capture it. The definition of the knowledge to be acquired can be assisted by classifying types of knowledge and knowledge sources (Dickinson 2000). To support this stage we propose to use an agent called a *Captor Agent*. The Captor Agent is responsible for collecting the information (data, models, experience, etc) from the different knowledge sources. It executes a proactive monitoring process to identify the information and experiences generated during the interaction between the user and the system or groupware tools (email, consulted web pages, chats, etc.). In order to accomplish this, the Captor Agent can use different techniques to acquire knowledge since there are several tools and techniques that consolidate and transform corporate data into information (Houari and Homayoun Far 2004). They contain:

- **Front-end system** (i.e. DSS-Decision Support System, EIS-Executive Information System and OLAP-Online analytical processing).

- **Back-end system**: Data warehouse, data mart, and data mining (Giannella, Bhargava et al. 2004).

Agents can also apply classical techniques used by experts to acquire knowledge such as: structured interviews, questionnaires,

![Figure 1. Knowledge Life Cycle Model Proposed](image-url)

*Formalizing / Storage*

*Acquisition*

*Evolution*

*Use*

*Transfer*
goal trees, decision networks, repertory grids, or conceptual maps (Rhem 2006). More sophisticated techniques such as webParser (Camacho, Aler et al. 2004) to obtain information from the Web, document classification (Novak, Wurst et al. 2003), mailing list management (Moreale and Watt 2003), or data mining and neuronal nets can be also used.

Once the knowledge has been obtained, the Captor Agent can classify it, by using ontologies, according to its type and the knowledge source from it was obtained (see Figure 2). This ontology is based on Rodriguez’s ontologies for representing knowledge topics and knowledge sources (Rodríguez, Martinez et al. 2004). The ontology has four knowledge source categories. These are: Documentation, which can be subdivided into: documentation related to the organization’s philosophy, documentation which describes the product/s which the company works with, documentation that describes the process that the company carries out, and other types of documentation that an organization has but that cannot be classified into any of the previous subgroups. Another important source where the Captor Agent finds information is the Web, which can also be divided into other subcategories such as Portals, Communities of Practice, etc. The main knowledge source in a company is, without any doubt, people. Depending on the type of company, people may be classified as clients, employees, etc. The last knowledge source that the Captor Agent can use is email that can be classified as internal

Figure 2. Knowledge Source Ontology
Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

mail (mail sent between employees), and external mail (emails sent to other people outside the organization).

One advantage of this approach is that the Captor Agent can work in any domain since by changing these ontologies the Captor knows what key knowledge should be found and where it might be.

b. Knowledge formalizing/storing is the stage that groups all the activities that focus on organizing, structuring, representing and codifying the knowledge with the purpose of facilitating its use (Davenport and Prusak 1998). To help carry out these tasks we propose a Constructor Agent. This agent is in charge of giving an appropriate electronic format to the experiences obtained so that they can be stored in a knowledge base to aid retrieval. Storing knowledge helps to reduce dependency on key employees because at least some of their expert knowledge has been retained or made explicit. In addition, when knowledge is stored, it is made available to all employees, providing them with a reference as to how processes must be performed, and how they have been performed in the past. Moreover, the Constructor Agent compares the new information with old knowledge that has been stored previously and decides whether to delete it and add new knowledge or to combine both of them. In this way, the combination process of the SECI (proposed in (Nonaka 1994)) model is carried out, producing new knowledge resulting in the merging of explicit knowledge plus new explicit knowledge.

Different techniques exist to store knowledge and frequently the technique used is narrowly related to the retrieval method used. Therefore, if a case-based reasoning is going to be used the knowledge will be stored as “cases”. Other techniques are knowledge objects, frames, predicate logic or fuzzy logic. In the case of using ontologies to classify the knowledge, methodologies to represent the knowledge can be used. Examples of these methodologies are: Ontolingua (Gruber 1993) or REFSENO (Representation Formalism for Software Engineering Ontologies) (Tautz and Von Wangenheim 1998).

c. Knowledge use is one of the main stages, since knowledge is helpful when it is used and/or reused. The main enemy of knowledge reuse is ignorance. Employers often complain because employees do not consult knowledge sources and do not take advantage of the knowledge capital that the company has. KMS should offer the possibility of searching for information; they can even give recommendations or suggestions with the goal of helping users to perform their tasks by reusing lessons already learnt, as well as previous experiences. In our model the agent in charge of this activity is the Searcher Agent, which searches in the knowledge base for the needed knowledge. Different techniques are currently used to search for knowledge. Many of them are based on the use of the position and frequency of keywords (Mohammadian and Jentzsch 2004) or on information retrieval techniques (Frakes and Baeza-Yates 1992; Liang and Huang 2000). Other authors such as (Sung Kim 2004) mix several techniques: data mining and case-based reasoning to develop a recommender system.

d. Knowledge transfer is the most investigated stage in knowledge management (Peachey, Hall et al. 2005). This stage is in charge of transferring tacit and explicit knowledge. Tacit knowledge can be transferred if it has been previously stored in shared means, for example: repositories, organizational memories, databases, etc. The transfer stage can be carried out by using mechanisms to inform people about the new knowledge that
Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

has been added. For this stage we propose a Disseminator Agent, which must detect the group of people, or communities who generate and use similar information: for example, in the software domain, the people who maintain the same product or those who use the same programming language. Therefore, this agent fosters the idea of a community of practice in which each person shares knowledge and learns thanks to the knowledge of the other community members (Wenger 1998). An appropriate knowledge management linked to communities of practice helps to improve the organization’s performance (Lesser and Storck 2001). Disseminated information may be of different types; it may be information linked to the company’s philosophy or specific information about a determined process. Finally, the Disseminator Agent needs to know exactly what kind of work each member of the organization is in charge of and the knowledge flows linked to their jobs. In order to do this, the Disseminator Agent contacts with a new type of agent called the Personal Agent which is in charge of determining the users’ profiles (it will be described in next section). Comparing this stage with the SECI model we can say that the Disseminator Agent fosters the socialization process since it puts people who demand similar knowledge in touch and once in contact they can share their experience, thus increasing their tacit knowledge.

e. Knowledge Evolution. This stage is responsible for monitoring the knowledge that evolves daily. To carry out this activity we propose a Maintenance Agent. The main purpose of this agent is to keep the knowledge stored in the knowledge base updated. Therefore, information that is not often used is considered by the Maintenance Agent as information to be possibly eliminated.

MULTI-AGENTS AGENCIES

Once the model and the agents that we propose to give support to the different stages have been described we are going to explain how the agents are structured into two agencies. Therefore, we group all the agents closely in charge of managing knowledge and supporting the different stages of the model proposed in one agency. Auxiliary agents are in another agency (see Figure 3).

Therefore, the Knowledge Agency is in charge of giving support to the KM process. It consists of the Constructor Agent, the Captor Agent, the Searcher Agent, the Disseminator Agent and the Maintenance Agent.

On the other hand, the User Agency is formed of the Personal Agent and the Interface Agent. The Personal Agent monitors users’ tasks to obtain their preferences and needs. In order to implement the Personal Agent user modeling techniques can be used. User modeling implies obtaining certain knowledge about the user. This knowledge describes what the user “likes” or what the user “knows” (Chin 1986).

The Interface Agent is the mediator between the users and the agents. Thus, when an agent wants to communicate a message to the user, the agent sends the message to the Interface Agent which shows it to the user.

Another component is the Shared Ontology which provides a conceptualization of the knowledge domain. The Shared Ontology is used for the consistent communication of the agencies.

In order to carry out the analysis and design of the agents involved we have followed a methodology called INGENIAS (Pavón and Gómez-Sanz 2003) which provides meta-models to define Multi-agent Systems, and support tools to generate them. Using meta-models facilitates the development of systems enormously, since they are oriented towards visual representations of concrete aspects of the system.
Below, we are going to show the different agent meta-model diagrams which describe the roles and tasks of each agent.

Figure 4 shows that the goal of the Captor Agent is to obtain information that should be stored. Its role is “Filter” since it must decide what information should be transformed into knowledge, the purpose being to use this in future projects. In the following lines, we describe each of the tasks carried out by this agent.

- **IdentifyIS**: This task consists of identifying available knowledge sources in the system.
- **CaptureInfo**: The agent must also capture information.
- **SendToConstructor**: Once the suitability of storing the information has been analyzed, the Captor sends it to the Constructor Agent (described in Figure 5) whose roles are: Sculptor and Treasurer since it is in charge of giving an appropriate electronic format to the information (Sculptor) and of storing it in the knowledge base (Treasurer). The tasks developed by Constructor Agent are:
  - **CompareInfo**: The agent is in charge of comparing the new information with the previously stored knowledge.
  - **CombineInfo**: The agent is also in charge of combining the new information with the previously stored knowledge.
  - **ClassifyInformation**: Another task is to classify the information received by the Captor Agent (for instance: models, structures, files, diagrams, etc.).
  - **SendToDisseminator**: This is a critical task which consists of sending knowledge to the Disseminator Agent.
  - **SaveKnowledge**: One of the most important tasks is to store the new knowledge into the knowledge base.
Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

The Disseminator Agent, whose role is PostOfficeEmpleee, as it behaves like the “postman” of the architecture, (see Figure 6) is composed of the next tasks:

- **SaveInfoTemp**: The Disseminator Agent stores temporarily the new knowledge received by the Constructor Agent.
- **EvaluateProfiles**: Once identified one user profile, the Disseminator Agent evaluates it in order to determine user’s needs.
- **LookForActivePersonalAgents**: Personal Agents can be distributed into different nodes, so it must identify all active Personal Agents available in the system.
- **SendInformation**: This is a critical task which consists of distributing the information to those people that can need it (really, the information is sent to their interface agents).
- **EvaluateInfo**: This task is focused on evaluating received information to be able to relate it with different user’s profiles.

Another agent that supports the knowledge life cycle is the Searcher Agent. The goal of this agent is to foster the internalization process of the SECI model, since the employees have the opportunity of acquiring new knowledge by using the information that this agent suggests. The
Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

Figure 6. Disseminator Agent diagram

Figure 7. Searcher Agent diagram

Figure 8. Maintenance Agent diagram
Searcher Agent diagram (Figure 7) is composed of the next tasks:

- **LookForInfo:** This agent is in charge of searching the information required by the users.
- **ClassifyInfo:** This agent also classifies the information found in the knowledge base.
- **SendInfoToI:** Finally, the agent sends the knowledge found in the knowledge base to a Interface Agent.

Last type of agent of the Knowledge Agency is the Maintenance Agent (Figure 8). The main purpose of this agent is to keep the knowledge stored in the knowledge base updated. Therefore, its task dealt mainly with deleting obsolete information.

Now, the two types of agents of the User agency are described. Figure 9 shows the Personal Agent diagram whose role is called “spy” since the agent must monitor users’ activities in order to obtain their profiles. Therefore, its goals are: monitoring users’ tasks and recommending information.

In order to attain these goals it should carry out the following tasks:

- **Modeling the users’ profiles:** By observing the users’ preferences, activities, information consulted, etc.
- **CreateManageLocalKnowledgeBase:** Creating and managing a “local knowledge base” where the relevant information for the user can be stored.
- **Recommending knowledge or knowledge sources:** This agent tries to guess what

---

**Figure 9. Personal Agent diagram**

![Personal Agent diagram](image)

**Figure 10. Interface Agent diagram**

![Interface Agent diagram](image)
knowledge would be relevant for the user. To accomplish this, this agent communicates with the Searcher Agent and with the Interface agent.

On the other hand, the Interface Agent is an intermediary between the users and the rest of agents, Figure 10 shows that its main tasks are: creating GUI and showing information to the users.

These tasks are defined in order to attain the goal of showing important information to the user, named in the diagram ShowInformation, so we have to create an user interface and put the received information from others agents in a nice way to the user.

**AGENTS COLLABORATION**

As it was mentioned before, the agents must collaborate with other agents. In order to show an example of this collaboration we are going to describe a possible scenario that can take place in an organization.

**Scenario**

Let us imagine that a person is writing a mail and the agents start to work in order to check whether the mail contains information that should be stored in the data base (we suppose that the employees know that the mail are reviewed and they agree with this).

As Figure 11 shows, the Interface Agent captures each event that is triggered by the Employee. In this case the employee sends an email. Then,
the Interface Agent warns the Captor Agent that an even has been triggered. Afterwards, the Cap-
tor Agent determines the type of groupware tool used (email) to identify and obtain information
topics about related task. In order to obtaining
information from the mail, a new agent can be
added to the system (it would not form part of our
architecture) but would be an agent that has been
already developed to assist in this task. There ex-
ist several agents implemented to deal with email
(Segal and Kephart 1999). Most of the current
implementations are text classifiers (Takkinen,
and Shahmehri, 1998) or keyword extractors
(Mock 2001). The Captor Agent would study
whether the information sent by the “email agent”
should be transformed into knowledge. Finally,
the Constructor Agent receives the information
which is structured in form of, for instance, cases
for its later storage.

RELATED WORK

Traditional KM systems have received certain
criticism, since employees are often overloaded
with extra work, as they have to introduce infor-
mation into the KMS and worry about updating
this information. One proposal to avoid this extra
burden was to add software agents to perform this
task in place of the employees. Later, intelligent
agent technology was also applied to other dif-
ferent activities, bringing several benefits to the
knowledge management process.

The benefits of applying agent technology
to knowledge management include distributed
system architecture, easy interaction, resource
management, reactivity to changes, interoperation
between heterogeneous systems, and intelligent
decision making. The set of knowledge manage-
ment tasks or applications in which an agent can
assist is very wide, for instance:

- To manage organizational memory, an ex-
  ample being the CoMMA project, (Gandon
  2000) (Corporate Memory Management
  through Agents), which combines emergent
technologies, allowing users to exploit an
organizational memory.
- To support cooperative activities. For in-
  stance in (Wang, Reidar et al. 1999) the
  authors propose a multi-agent architecture to
  provide support to cooperative activities.
- To recommend. For instance in (Sung Kim
  2004) a system to customize recommendations
  is described.
- To find experts. Some systems are used to
  help people find experts which/who can
  assist them in their daily work.
- To share knowledge. For instance in (Mercer
  and Greenberg 2001) a multi-agent system
  is proposed for knowledge sharing in a sys-
  tem designed to advise good programming
  practice.
- To manage mailing lists, or document clas-
  sification (Moreale and Watt 2003).

These and other existing systems were often
developed without considering how knowledge
flows and what stages may foster these flows.
Because of this, they often support only one
knowledge task, without taking into account that
knowledge management implies giving support
to different process and activities. On the other
hand, KM systems often focus on the technology,
without taking into account fundamental problems
that these kinds of systems are likely to support
(Hahn and Subramani 2000).

CONCLUSION AND FUTURE WORK

The main contributions of this paper are the design
of knowledge cycle for developing KMS where
the main functions that this kind of systems must
support are described. Moreover, a multi-agent
architecture is outlined to help KMS developers to
implement these kinds of systems. The advantages
of these contributions are:
The model provides support to different activities: knowledge creation, storage/retrieval, transfer and application. All are activities which, according to the authors who specialize in evaluating KMS, should support this kind of system.

The architecture is based on a KM life cycle that we have proposed for this end. Therefore, we try to avoid the lack of other architectures that are focused on the technology and forget the knowledge aspects.

The architecture makes use of intelligent agents. This is a technique that have proved to be very convenient in knowledge management activities since it avoid one of the problems of some KMS such as overloading the employees with extra work instead of helping them during their daily work. Agents can carry out many tasks on behalf of users. Moreover, they act when they consider that it is necessary to do so without needing users’ instructions. Another advantage of using agents is that they can collaborate with other agents already implemented to carry out concrete knowledge tasks. For instance obtaining information from the Internet or from e-mail. Thus, the development of KMS would be easier since only the basic agents of our model would have to be implemented and these could collaborate with other agents that have already been tested.

On the other hand, we are modelling the agents in a systematic way by using INGENIAR methodology whose meta-models help future developers to understand how the different agents work.

As future work we aim to compare the implementation of a KMS based on our proposal with developments using other architectures. Without any doubt this evaluation will help us to improve our proposal. On the other hand, we are also working on extending the model documentation with a more wide and detailed description of the possible techniques that could be used to implement each type of agent according to the main needs that organizations usually demand.

From a technological point of view, we are also studying JADE in order to see how easy it would be to migrate to this new platform. The current prototype was implemented using JADE (Java Agent Development Framework) since it is FIPA compliant and is currently one of the most widely used. Moreover, JADE has been successfully used in the development of other systems in the domain of knowledge management (Bergenti, Poggi et al. 2000; Gandon 2000).

ACKNOWLEDGMENT

An earlier version of this paper was presented at the 40th Hawaii International Conference on System Sciences (HICSS-2007), 03-06 January 2007, Waikoloa, Big Island, Hawaii.

This work is partially supported by MELISA project (PAC08-0142-3315) and MECENAS (PBI06-0024) project, both of them conceded by Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain. This is also supported by the ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain.

REFERENCES


Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

Memory Management Systems. ECAI’2000, Workshop on Knowledge Management and Organizational Memories.


Helping to Develop Knowledge Management Systems by Using a Multi-Agent Approach

GIR Conference on Research and Development in Information Retrieval (pp. 392-393). New Orleans, Louisiana, USA.


van Elst, L., V. Dignum, et al. (2003). Agent-Medi
diated Knowledge Management. *Agent-Medi
dated Knowledge Management*, Springer.

Architecture for Cooperative Software Engineer-
ing*. Paper presented at the Eleventh International
Conference on Software Engineering and Knowl
dge Engineering (SEKE’99), Germany.

Ward, J. and A. Aurum (2004). *Knowledge Man-
gagement in Software Engineering - De-scribing
the Process*. Paper presented at the15th Aus-
tralian Software Engineering Conference (ASWEC
2004), Melbourne, Australia, EEE Computer
Society Press.

Wenger, E. (1998). *Communities of Practice:
Learning, Meaning, and Identity*. Cambridge,
U.K.: Cambridge University Press.

Wiig, K. M. (1997). Knowledge Management:
Where did it come from and wher will it go?

Agents: Theory and Practice. *The Knowledge
Engineering Review, 10*(2), 115-152.