Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings
Americas Conference on Information Systems (AMCIS)

2005

Technological Support for a Learning Oriented Knowledge Management System

Dianne Hall
Auburn University, halldia@auburn.edu

Yi M. Guo
University of Michigan - Dearborn, magyiguo@umd.umich.edu

Casey Cegielski
Auburn University, cegieca@auburn.edu

Robert A. Davis
Texas State University - San Marcos, rd23@business.txstate.edu

Technological Support for a Learning-Oriented Knowledge Management System

Dianne J. Hall
Auburn University
dhall@auburn.edu

Yi Guo
University of Michigan - Dearborn
magyiguo@umd.umich.edu

Casey Cegielski
Auburn University
cegieca@auburn.edu

Robert A. Davis
Texas State University – San Marcos
rd23@business.txstate.edu

ABSTRACT

Knowledge management is quickly becoming a requirement for today's complex organizations. Creating and managing existing knowledge has been linked to successful innovation and to sustainable competitive advantage. However, systems specifically designed to manage knowledge, support knowledge creation, and verify existing knowledge are in their infancy. This article follows the framework for a Learning-Oriented Knowledge Management System, and shows how such a complex system can be supported by an equally complex technology – that is, a multi-agent system.

We define single agents and multi-agent systems and subsystems in the context of knowledge management systems in general, and the Learning-Oriented Knowledge Management System (LOKMS) specifically. We show how a multi-agent system can be conceived to fully support the LOKMS, describe some necessary agents and agent subsystems, and demonstrate prototypically a multi-agent system designed and built to support the integrity-checking component of the LOKMS. This system begins the process of LOKMS design and development.

Keywords

Knowledge management, knowledge management systems, Churchmanian systems, agent technology.

INTRODUCTION

Knowledge management (KM) is a term that is in the forefront of managerial and technological issues. Today's organizations operate in complex environments wherein functioning effectively requires the ability to access more relevant and accurate information in decreasing amounts of time. To effectively manage these tasks, organizations must adopt flexible technologies that function well in dynamic environments and which enable the organization to evolve and maintain a reliable data store.

A framework for such a system was developed by Hall, Paradice, and Courtney (2003). They use the philosophical foundations of Churchman's (1971)inquiring systems and the concept of an inquiring organization (Courtney, Croasdell and Paradice, 1998) to conceptualize a KM system capable of functioning in complex environments. At the same time, it supports many requirements of an organization's KM initiatives such as creating an environment in which acquisition, storage, and retrieval of information supports decision-making, action, and learning.

The sheer complexity of the task of implementing a comprehensive KM system requires the support of a correspondingly complex technology. Hall et al. (2003) suggest that a future research direction is to investigate technology that can support their system. We begin that endeavor herein and maintain that agent technology may provide the basis for the learning-oriented KM system (LOKMS) conceptualized by them.

THE LEARNING-ORIENTED KNOWLEDGE MANAGEMENT SYSTEM

Foremost in the design of the LOKMS are components founded on the theory of inquiring systems (Churchman, 1971). Churchman defines different modes of inquiry based on the philosophical foundations of five Western philosophers, for whom each inquirer is named. Each of the inquirers seeks to generate new knowledge based on incoming information and functions more effectively in particular contexts. For instance, the Leibnizian inquirer is appropriate for structured problems that have a definite solution and allow for analytical formulation such as personnel scheduling, whereas the Lockean inquirer is appropriate for structured problems that lend themselves to consensus such as formulation of a 5-year plan. The Kantian inquirer is most appropriate for problems that are moderately unstructured and frequently described by multiple perspectives, but which allow for analytical formulation such as a budget process. The Hegelian inquirer provides support for problems that are divisive in nature and possibly ill structured. Common scenarios for the Hegelian inquirer are similar to those of negotiation such as labor/management disputes. The Singerian inquirer is capable of handling all problem structures and all scenarios. It is based on the theory of socially constructed information with particular attention paid to the perspectives of all stakeholders and of the outcome's effect on each. Such a complex inquirer requires extensive resources and generally would not be invoked if a less complex inquirer was also appropriate.

Although Churchman (1971) discussed these various inquirers, he also discussed several requirements from which components may be derived to support a decision-making or knowledge creation process. In fact, the LOKMS diagram (see Figure 1) does not emphasize the inquirers (which are located in the Analysis Module of the Knowledge Creation Unit), but rather the *process* of inquiry, including integrity-checking, feedback, information discovery, hypothesis generation, and environmental scanning. These modules and supporting components are centered on organizational memory which provides its critical foundation (the Knowledge Storage Unit); both the Information Gathering Unit and the Knowledge Creation Unit interact with the Knowledge Storage Unit and with each other through the Knowledge Storage Unit.

We maintain that, because the LOKMS positions organizational memory as its core, a first attempt to design technological

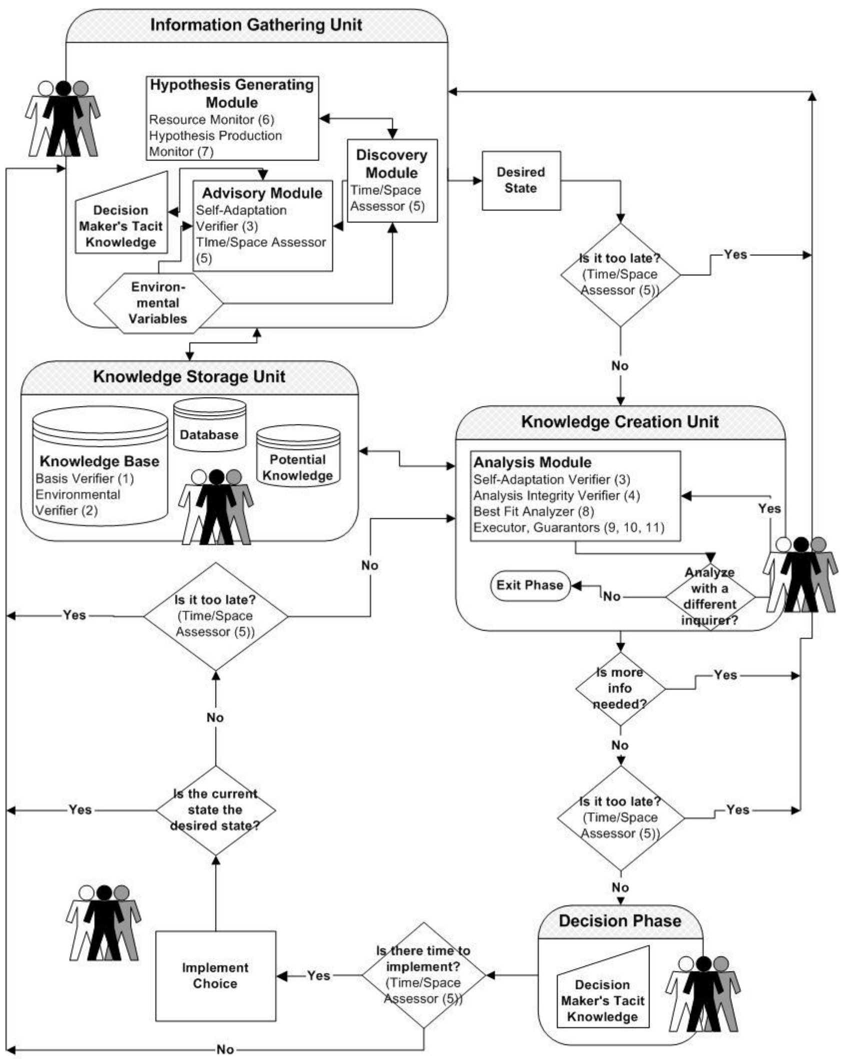

Figure 1. Learning-Oriented Knowledge Management System (LOKMS) (Hall et al., 2003)

support for a system should be concerned with processes to ensure the integrity of the artifacts therein. We are concerned here with how supportive technology may be used to implement the specific supporting components conceptualized by Hall, et al. (2003) that ensure the integrity of items in organizational memory. These components comprise what we will call the integrity-checking process that allows a learning-oriented organization to expand its organizational memory with accurate, timely knowledge and on which our prototypical system is based. Table 1 summarizes these integrity-checking components.

Integrity-Checking Component	# on Fig. 1	Critical Requirements	Techniques
Basis Verifier	1	Reviews currently stored knowledge for inconsistencies or undetermined relationships	Truth maintenance
Environmental Verifier	2	Reviews incoming information for impact on currently stored knowledge	Perception, learning, and mobility
Self-adaptation Verifier	3	Identifies new knowledge or relationships based on new information and prepares action reports as necessary	Default reasoning, learning
Analysis Integrity Verifier	4	Warns user of potential problems with new information	Proof and explanation
Time/Space Assessor	5	Performs time/space assessment	Temporal logic

Table 1. Integrity-Checking Components of the LOKMS

A *basis verifier* is critical to inquiring systems design, as an inquiring system such as the LOKMS depends on its knowledge bases to create new knowledge and analyze information. If any of the information in a knowledge base is incorrect, the system may reach the wrong conclusion about new items. An *environmental verifier* varies from a basis verifier in that, rather than continuously comparing knowledge in the store against itself, it reviews incoming information or newly created knowledge to determine whether changes have occurred in the environment that would, in turn, affect the validity of other stored knowledge components.

Commonly, knowledge becomes obsolete as an organization's processes or resources (particularly technology) mature. A *self-adaptation verifier* allows the system to support management by preparing reports of recommended action in the face of new knowledge, especially in response to changing environmental variables. This component also monitors organizational memory changes to identify new relationships or new knowledge that arises from newly stored knowledge or changing information. A component such as an *analysis integrity verifier* can warn a user of potential problems with information or knowledge that the user wants to store, while the primary functions of the *time/space assessor* are to follow time-critical missions of the organization, to ensure that all temporal considerations of the organization are met, and to maintain time-sensitive knowledge and information such that outdated information is removed and information nearing its expiration is flagged and possibly sent to the self-adaptation verifier for action. Together, these five components comprise the integrity process that provides the organization with reliable organizational memory, allowing both the LOKMS and organizational members to securely engage in the knowledge creation process.

A complex system such as the LOKMS has a long list of potential system requirements such as continuous information gathering (discovery). Without examining both that which is "known" to the organization and that which is new to the organization, challenges to existing assumptions and knowledge will not occur, and an opportunity for knowledge creation may be lost. This element is inconsequential if the knowledge within the system has not undergone diligent integrity checking both as environmental conditions change and as new information and knowledge are assimilated into the system. This requires the ability to recognize a changing environment and respond accordingly as well as warn the user against storage of information that potentially or directly conflicts with information in the knowledge base. Thus, integrity checking is also a crucial characteristic that must be supported during design of a learning-oriented organization support system, particularly in conjunction with new information discovery. One must also remain aware that the human element of the organization requires support. Rarely are knowledge creation and management tasks that are confined to a single individual. Often it is a complex, highly social task involving many individuals, organizational units, or supply chain partners. Thus, another requirement of a complex system may be the ability to abide by social interaction requirements such as cooperation, negotiation, and coordination.

The combination of dynamic requirements (e.g., discovery, integrity-checking) and social requirements (e.g., cooperation) requires a technology that is capable of both procedural and social tasks. We believe that agent technology is an appropriate technological tool to use to conceptualize, design, and implement such a system. However, before we can begin the discussion of a multi-agent system designed to support the LOKMS, we must briefly introduce agent technology and its particular applicability to such a complex system.

AGENT TECHNOLOGY

A software agent (hereafter referred to simply as an agent) is a self-contained program capable of controlling its own decision-making and acting, based on its perception of its environment, in pursuit of one or more objectives (Jennings and Wooldridge, 1996). In particular, agents enjoy properties of autonomy, responsiveness, pro-activeness, and social ability. Two key distinguishing characteristics of agents are that they 1) are capable of handling relatively high-level tasks, and 2) exist in an environment that may dynamically affect their problem-solving behavior and strategy. Of particular importance in agent development is the Belief-Desire-Intention (BDI) model (Rao and Georgeff, 1995) wherein an agent acts to achieve a goal (desire) by performing specific acts (intentions). Desires and intentions are based on the agent's underlying beliefs, which are similar to the non-agent usage of the word in that those beliefs affect how the agent perceives or acquires information. Although these beliefs may be updated as the environment changes, the foundation of the beliefs remains reasonably constant.

The BDI model has been applied in various contexts; notably, its applicability to a multi-perspective decision system has been demonstrated (Hall, Guo, Davis and Cegielski, 2004). Their paper extends the Unbounded Systems Thinking (UST) model (Mitroff and Linstone, 1993) with the BDI model. In their demonstrative system, agents were created that represented individual stakeholders and their individual value system and characteristics that paralleled the technical, organizational, and personal perspectives of the UST model. Their simulation indicated that the demonstrative system performed the same task as its human counterparts, arrived at a similar conclusion, but introduced less subjectivism in the decision.

There are several other recent applications of agent technology to information systems (e.g., Aguirre, Brena and Cantu, 2001; Lee and Chong, 2003; Roda, Angehrn, Nabeth and Razmerita, 2003). However, in these efforts, there is no explicit support for the integrity-checking components of the LOKMS. We suggest the following framework as a multi-agent system to close this gap.

A PROTOTYPICAL MULTI-AGENT COOPERATIVE SYSTEM

A knowledge management system must manage but also expand organizational memory through organizational learning which increases an organization's ability to act effectively in given situations (e.g., Alavi, 2000; Davenport and Prusak, 1998; Grant, 1996). The architecture on which to base this process should include features for facilitating information/knowledge acquisition, discovery, and sharing as well as supporting codification, storage, and management of explicit knowledge. Through detailed explanation of the prototype below, we emphasize elements of the LOKMS that verify integrity, promote discovery, suggest action, and provide inquiry. In doing so, we provide the basis for organizational learning and knowledge management in the spirit of the LOKMS.

The collective goal of the LOKMS agent system is to provide relevant, accurate information by maintaining a consistent knowledge base and by providing organizational strategies for processes such as decision-making or goal setting. Each organization, or unit within an organization, may choose to structure the system differently to fit its specific needs. One of the many assets of agent technology is the above ability to move, multiply, and restructure its agent configurations as required by any given context. Unlike other technologies, which, once established, remain fairly static (for instance, a database), agent-based technologies are free flowing and dynamic. Thus, single agents or agent systems can be designed to handle specific tasks. Although for simplicity we focus in this work on a minimal system, it is possible for an agent-based system to contain hundreds of agents with the ability to work with, or on behalf of, the others. Each of the agents in the prototypical system have distinct tasks but work together to achieve the goal of knowledge creation and management within the organization. Figure 2 shows the diagrammatic flow of the system; details will be presented with the prototypical system explanation. Organizational memory is shown here as centralized; this is for illustration simplicity only. The knowledge base can be centralized or distributed over the organization; in addition, any agent may have local memory to facilitate processing.

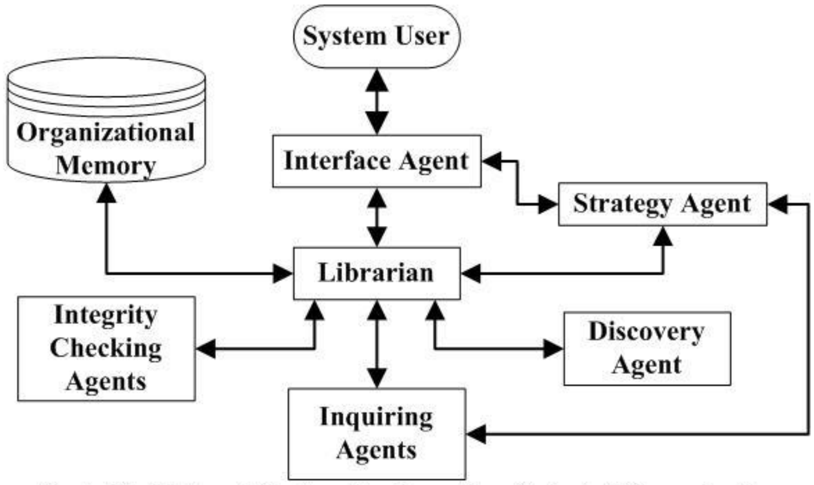

Figure 2. A Multi-Agent System for Learning-Oriented Organizations

Although all of the features of the LOKMS are important and have specific and vital roles within a learning-oriented organization multi-agent system, it is the integrity-checking feature that is perhaps the most vital, particularly in the spirit of Churchman's (1971) inquiring systems. It is because of this importance that we use it to design and build a multi-agent system to illustrate that the multi-agent system conceptualized herein is feasible. The prototype demonstrates the use of multiple agents in a simple scenario to prevent erroneous sales promotions from being entered into the system and to suggest new sales promotions.

The system was built using Java-based open-source software that is freely available and has on-line support in the form of discussion forums as well as support documentation. The primary software used for this demonstration is Jadex, which was chosen specifically because of its close association with the popular Belief-Desire-Intention (BDI) model of agent technology (Rao and Georgeff, 1995) used in the multiple perspective work of Hall and colleagues (2004). The Jadex agent system is a BDI extension for the JADE agent platform (Pokahr, Braubach and Lamersdorf, 2003). It focuses on goal-directed behavior and allows the explicit declaration and usage of goals (Braubach, Pokahr, Moldt and Lamersdorf, 2004). Another important motivation for the realization of Jadex is its usability in mainstream software engineering by extensively employing well-established techniques like Java, XML, and OQL.

The premise of the example system is simple. A retail outlet has many promotions in place at any one time. Each store manager is able to introduce new promotions within constraints set by the corporate office. These constraints are the number of the same product that must be purchased to qualify for the promotion, the amount of the discount to be applied, and the length of time the promotion can run. These constraints vary from product to product. Before the manager can begin the new promotion, it must be entered into the database. This process requires that the input system check for valid promotions. These checks include verifying a legitimate product number as well as the constraints on each product. Each promotion entered is first verified for technical accuracy of the input (i.e., data validation), followed by verification of a valid product number. Once those checks are passed, the promotion is then checked for reasonableness (number to purchase and discount amount) and the timeliness of the promotion. If a given store has not initiated a promotion in 30 days, the system is initiated to suggested sales promotion possibilities. This understandable yet simple situation is complex enough to require several agents.

The Interface Agent is responsible for constructing and maintaining the user interface and the tasks associated with data validation. Additionally, this agent acts as a coordinating agent. When the system is activated, the Interface Agent activates all the supporting agents before presenting the GUI to the user. This prevents the user from attempting an input process prior to the integrity-checking agents being available. In addition, the Interface Agent provides centralized communication by eliciting responses from each agent in turn. Although agents can and do communicate with each other, this particular arrangement more closely controls the integrity-checking process and allows the Interface Agent to pass queries from agents to the user as needed (for instance, if error checking shows a problem).

Once the field validation process is complete, the Interface Agent sends a query to the Product Agent. This agent represents the *basis verifier*. This agent keeps the integrity of the system intact at the most basic level by confirming the legitimacy of a product for which a promotion is requested. The Product Agent will return either an error message or a confirmatory message indicating a legitimate product to the Interface Agent, who then communicates with the user to rectify the problem or continues the integrity-checking process.

After successful product verification, the Interface Agent requests a second integrity check to determine whether the promotion (or a similar one) is currently in place in that store. The Sales Promotion Agent is responsible for all aspects of the sales promotion database, including tasks such as preventing redundant promotions. While performing this task, the Sales Promotion Agent functions as an *analysis integrity verifier* by warning the user of duplicate or current promotions involving the same product, thus preventing redundant promotions from being introduced. This agent is also involved later in the process when a newly approved promotion must be entered into memory at which point it is functioning as a librarian by controlling access to organizational memory. This becomes central to the system because each of the processes active within the system will, at some point, interact with the information contained in organizational memory. Allowing only one agent (or agent system) to control access provides control for allocating resources for organizational memory. This agent may also interact with other agents in the system, such as requesting clarification of information requests received from the interface agent or additional information from the discovery agent system.

After the Sales Promotion Agent verifies that the incoming promotion is unique, the Interface Agent will invoke two agents to verify that the proposed promotion is within corporate guidelines. Analysis of the propriety of the quantity and discount amounts is conducted by the Constraint Agent, and analysis of the length of time is conducted by the Time Assessor Agent. Both of these agents function as *analysis integrity verifiers* by warning the user of constraint violations; separation of duties allows the unique task of date analysis to be performed by one agent that also follows other date issues. Together, they verify reasonableness of the requested promotion.

When the above checks have been satisfactorily concluded, the Interface Agent first requests that the Sales Promotion Agent accept the promotion as valid and record it in the sales promotion information base, and then informs the user that the promotion has been accepted.

Although the prototype was designed primarily to exemplify the integrity-checking process, an additional feature is required to demonstrate how the inquiry agent system interacts with the other agents. Using a temporal perspective, the prototype assumes that promotional inactivity of a given store for 30 days triggers the Sales Promotion Agent to request sales promotion suggestions from the Strategy Agent. The Strategy Agent's function is to differentiate between the five inquirers and their representative agents based on the type of problem being considered. It investigates the problem, determines that it is appropriate for the Leibnizian Inquiring Agents because it is structured, and passes the request to that agent. This agent may also examine the results of the analysis, call on a different inquirer if appropriate, and interact with the Sales Promotion and Interface Agents as necessary to facilitate information flow. In addition, this agent functions as a decision support or expert system support component for the user when required by providing problem-solving strategies to the user.

Once the Leibnizian Agent receives the request from the Strategy Agent, it examines the problem using historical information (e.g., past promotional events), information acquired from the Discovery Agent through the Sales Promotion Agent (e.g., competitors' current promotions), and statistical analysis (e.g., cost/benefit analysis) to arrive at a suggestion that includes product, length of promotion, and discount ranges. This suggestion is passed back to the Strategy Agent, which then passes the suggestion on to the Interface Agent to inform the store manager who may choose to act on the suggestion, request additional suggestions, or continue with no promotions.

Action suggestions based on promotional inactivity are not the only triggers that may occur in the system. For instance, a deletion from the product list should initiate a review of current promotions to see whether a promotion exists on a now obsolete product, requiring a warning to be issued to the user. An addition to the product list may trigger a suggested promotion based on the new product. Changes to the quantity, discount, and/or length constraints should trigger analysis of current promotions to verify that the new changes do not invalidate existing promotions. Passage of time may trigger analysis of current promotions and generate a warning regarding those promotions that are soon to expire. A continuous stream of promotions on one or a small set of products may trigger a suggestion to broaden the scope of promotions.

Changes to products or constraints require that the beliefs of the associated agents also change. For instance, because the Sales Promotion Agent is the one that maintains the information base on promotions, it must be informed when changes have been made to the product database, which also contains appropriate promotion constraints. Thus, a Change Agent is developed to function as a self-adaptation verifier. The task of this agent is to monitor user change requests and inform the appropriate agents of the changes, triggering those agents to adjust their beliefs accordingly. Belief adjustment may simply require an insertion or deletion (in the case of a product change) or an update (in the case of a constraint change).

Changes to promotion constraints will trigger the Sales Promotion Agent to invoke the *environmental verification* function to ensure that the changes did not invalidate any promotions. If a problem exists, the Sales Promotion Agent will request that the Interface Agent issue an exception report for user action. The Change Agent may also monitor organizational memory changes to identify new relationships or new knowledge that arises from newly stored knowledge or changing information, such as promotions that are successful and those that are not.

Many of these triggers are set in motion with cooperation of the Sales Promotion and Strategy Agents calling on the Inquiring Agents and the Discovery Agent as needed. The task of the Discovery Agent (to both support organizational learning and information discovery and acquisition) is to routinely scan the external environment of the organization for changes that warrant consideration or action. Like the Intelligence Gathering Unit in Hall et al.'s (2003) model, this agent is yet another multi-agent system that may contain, in addition to discovery capabilities, the ability to hypothesize regarding the relevance and/or urgency of incoming information or the ability to advise the user (via the Librarian and Interface Agents) when changes are observed. It may also be invoked by the Librarian Agent to respond to a user query for new information, or to provide additional information to the inquirers.

Each time a promotion is suggested and that promotion is subsequently executed and analyzed for its effectiveness, more is understood about the nature of promotions and in turn, the organization has learned. Table 2 summarizes the agents used in the prototype.

Agent (Module from Figure 2)	LOKMS Component Represented	Belief(s)	Interacts with
Interface	None	Directory of users, user's unique requirements	User, sales promotion agent
Sales Promotion (Librarian)	Analysis integrity verifier, environmental verifier	Current and former sales promotions	All agents
Product (Integrity Checking)	Basis verifier	Current product list	Sales promotion agent, all integrity-checking agents
Change (Integrity Checking)	Self-adaptation verifier	Current product list, current constraints	Sales promotion agent, all integrity-checking agents
Constraint (Integrity Checking)	Analysis integrity verifier, environmental verifier	Current quantity and discount constraints	Sales promotion agent, all integrity-checking agents
Time Assessor (Integrity Checking)	Analysis integrity verifier, time/space assessor	Current time constraints, current date	Sales promotion agent, all integrity-checking agents
Strategy	Advisory Module	Range of acceptable profit margins per store	Interface agent, sales promotion agent, inquiring agents
Discovery	Discovery Module	Competitor positions, market positions, consumer preferences	Sales promotion agent
Leibnizian Inquirer	Analysis Module	Profit maximization, promotion effectiveness	Sales promotion agent, strategy agent

Table 2. Summary of the Prototypical Multi-Agent System

Individuals also play an indispensable role in these systems. Individuals provide the domain knowledge to be captured, and verify the correctness of its representation during system design, construction, and implementation. The elements and models inherent in the design and use of a system must be understandable by all stakeholders (Kilov, 2002) and therefore depend not only on the expertise of the designers, but the expertise and direction of the users. While agents may play a large role in producing relevant information, the system user should decide whether a piece of information is relevant or important to a specific context. Further, the function of choice (that is, the decision) resides with the organizational member.

Many avenues for future research exist. While we have shown that extant technology may be used to implement the system, it may be of interest to study working examples to highlight the possibilities and problems associated with those technologies. Parts of this system have been created and tested as new software using JADE agents, but further development and testing must be done. Technical, security, and usability issues can be identified and ways to mediate them tested.

Investigating the acceptability and impact of such a high-level technology in an existing organization may also be of interest. In some arenas, agents are regarded as somewhat mystical and perhaps magical – and thus unsuited to "average" processes within organizations. De-mystifying the technology is a critical step to enhancing its acceptance across various domains. Research should be designed to determine where potential adoption problems may exist and to determine who within an organization is most likely to support such a change. Once a system has been fully designed and implemented, concrete measures of organizational memory validity must be developed and tested within the confines of a particular context or domain. Feasibility of transfer across contexts or domains may then be the focus of new research.

CONCLUSION

In this paper, we develop a prototypical system to demonstrate that agent technology can be used to support integrity-checking and promotion suggestion. We believe this work can be extended to complete technological support of knowledge creation and management in a learning-oriented organization. The work of Hall, Paradice, and Courtney (2003) provides a basis for our conceptualization of the system and its components; agent technology is used as the technological design framework. The concept of agent technology has been used to conceptualize and model agents and multi-agent systems that serve to support the foundations of a LOKMS.

Although we focus on the technological design aspect of the Learning-Oriented Knowledge Management System (Hall et al., 2003), we also embrace Churchman's (1971) admonitions to maintain multiple perspectives, flexibility, learning, and social orientation by designing agents to represent different tasks and perspectives and strongly advocate and design for individuals in the system.

The multi-agent system conceptualized here has the ability to facilitate all components of a knowledge management system: knowledge creation, knowledge storage and retrieval, knowledge transfer, and knowledge application (Alavi and Leidner, 2001). These concepts are important to any organization desiring to exploit its knowledge resources, but are particularly suited to the pursuit of knowledge creation and management. We believe that this support system framework will lead to a sustainable materialization of and support for the emerging form of a learning-oriented organization.

REFERENCES

1. Aguirre, J. L., Brena, R. and Cantu, F. J. (2001) Multiagent-based Knowledge Networks, *Expert Systems with Applications* 20, 1, 65-75.
2. Alavi, M. (2000) Systems for Managing Organizational Knowledge, in *Framing the Domains of IT Management: Projecting the Future through the Past*, R. W. Zmud and M. F. Price (Eds.), Pinnflex Educational Resources, Cincinnati, OH, 15-28.
3. Alavi, M. and Leidner, D. (2001) Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues, *MIS Quarterly* 25, 1, 107-136.
4. Braubach, L., Pokahr, A., Moldt, D. and Lamersdorf, W. (2004) Goal Representation for BDI Agent Systems, in R. H. Bordini, M. Dastani, J. Dix and A. El Fallah-Seghrouchni (Eds.) *Proceedings of the Second International Workshop on Programming Multiagent Systems*, July 20, New York, NY, 9-20.
5. Churchman, C. W. (1971) The Design of Inquiring Systems: Basic Concepts of Systems and Organizations, Basic Books, Inc., New York, NY.
6. Courtney, J. F., Croasdell, D. and Paradice, D. B. (1998) Inquiring Organizations, *Australian Journal of Information Systems* 6, 1, 3-15.
7. Davenport, T. H. and Prusak, L. (1998) Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, Boston.
8. Grant, R. M. (1996) Toward a Knowledge-Based Theory of the Firm, *Strategic Management Journal* 17, Special Winter Issue, 109-122.
9. Hall, D. J., Guo, Y., Davis, R. and Cegielski, C. (2004) Extending Unbounded Systems Thinking with Agent-Oriented Modeling: Conceptualizing a Multiple-Perspective Decision-Making System, *Decision Support Systems*, forthcoming.
10. Hall, D. J., Paradice, D. B. and Courtney, J. F. (2003) Building a Theoretical Foundation for a Learning-Oriented Knowledge Management System, *Journal of Information Technology Theory and Application (JITTA)* 5, 2, 63-89.
11. Jennings, N. R. and Wooldridge, M. J. (1996) Software Agent, *IEEE Review* 42, 1, 17-20.

12. Kilov, H. (2002) Business Models: A Guide for Business and IT, Prentice-Hall, Inc., Upper Saddle River, NJ.
13. Lee, Y. and Chong, Q. (2003) Multi-agent Systems Support for Community-Based Learning, *Interacting with Computers* 15, 1, 33-55.
14. Mitroff, I. I. and Linstone, H. A. (1993) The Unbounded Mind: Breaking the Chains of Traditional Business Thinking, Oxford University Press, New York.
15. Pokahr, A., Braubach, L. and Lamersdorf, W. (2003) Jadex: Implementing a BDI-Infrastructure for JADE Agents, *EXP - In Search of Innovation* 3, 3, 76-85.
16. Rao, A. S. and Georgeff, M. P. (1995) BDI Agents: From Theory to Practice, in V. Lessor (Ed.) *Proceedings of the First International Conference on Multi-Agent Systems,* June 12 - 14, 1995, San Francisco, CA, 312-319.
17. Roda, C., Angehrn, A., Nabeth, T. and Razmerita, L. (2003) Using Conversational Agents to Support the Adoption of Knowledge Sharing Practices, *Interacting with Computers* 15, 1, 57-89.