

Theme 7: Intelligent project lifecycle knowledge management and decision support

Abstract

Rapid advances in technologies such as networks, database analysis techniques, and high speed processing helped the growing importance of organizational decision support systems (DSSs). Today's organizations invest in various knowledge management (KM) systems and tools to enable seamless integration of the constantly increasing volume and sources of information. However, with the myriad of commercially available DSSs, organizations are facing the problem of a growing gap between the solely available applications and the business users' requirements for effectively take advantages of their support in the decision making process. To narrow this gap, an integrated and flexible environment is needed to assimilate the use of many of these systems and satisfy the growing users and needs. In this paper, we argue that decision support systems need to use software agents as the natural means of representing the broadly categorized business users of an organization. They hide from their users the complexity of underlying technology of data extraction, analysis, information retrieval and knowledge sharing, and enabling them to make fact-based and timely decision with more degree of confidence. This research approach aims to address the research issue of how KM can be optimized using intelligent agents and how to enhance decision-making process. Second, we delineate the design and implementation of such environment, based on software agent development technology, and endowed with various types of data access capabilities and information retrieval. The system supports multiples business users. The proposed system is applied to a real-world project lifecycle case that is SE (Software Engineering) project. A prototype of the system is presented where intelligent agents are the building blocks of a peer-to-peer organization wide system. The application was implemented using Eclipse technology, and the agents were deployed on the FIPA-OS (Foundation for Intelligent Physical Agents-Open-Source) environment, we used JESS (Java Expert System Shell) to develop the knowledge based of the agents' reasoning.

1. Introduction

Decision support systems are of increasing importance for organizations survival, growth, and competence. They refer to computer-based systems that are developed to aid the process of decision-making, by supporting the solution of non-structured management problem [1], [3]. They are characterized by the following features: they incorporate data/models; they support not replace decision makers, and they target ill-structured problems. Given the dynamics of today's business environment, most organizations competitive survival and on going growth will depend on their ability on constantly understanding, tracking and adapting organizational goals and purposes. Although decision tools and decision support systems are subject to much research and industrial applications, they usually fall short because of their small-scale focus and can not simply scale up to cover broad organizational needs. With the

advent of software agent technology the development of broad-spectrum and complex systems are possible. We argue that decision support systems need to use software agents as the natural means of representing the broadly categorized business users of an organization. They hide from their users the complexity of underlying technology of data extraction, analysis, information retrieval and knowledge sharing, and enabling them to make fact-based and timely decision with more degree of confidence.

Although in the literature, the definition of agents is rather blurred and relatively dependent on who is designing and developing them, and for what purpose they are used for [2], most of the researchers in this area agree that software agent are: *autonomous* entities capable of exhibiting goal oriented behavior and take initiatives accordingly; *knowledgeable* in the sense that they are able to manage their own beliefs (information) and desires (goals) and they can decide upon the next operations to execute; *interactive* capable of interacting with the environment and communicating with other agents; and *adaptable*, can change their own behavior through their own learning, user customization or the download of new capabilities. From process viewpoint, agents are task specific entities assigned to a process to run as an executable code that has its data and state.

In this research we use agents to support distinct individuals (business users) of an organization in their decision making process. Each agent is an autonomous, an active problem solver entity with its own goals to achieve (which is usually delegated to from the business user), has control over its actions, uses its own knowledge capabilities, adapts, interacts, and responses to changes in its environments. The agents are provided with various types of data access capabilities to access and analyze the data contained in the organization's data warehouse (DW), on-line transaction processing (OLTP) and legacy systems, in order to retrieve the critical information needed by the decision makers.

The system is applied to a real world project lifecycle decision support for a SE (Software Engineering) project development.

The remainder of this paper is organized as follows. Section 2 illustrates the concept of knowledge management for decision support. Section 3 depicts the literature review and research methodology. In Section 4 we delineate the contribution and motivations of the proposed system. In Section 5 we describe the system architecture and structure. Section 6 details the system prototype and implementation. And finally conclusion and future work are given in Section 7.

2. Knowledge Management for Decision Support

2.1 Concepts

Information outsourcing is one of the major concerns of large companies. In recent years

advances in technologies such as networks and databases have enabled businesses to store enormous internal and external data to support decision-making. Today, any enterprise to be competitive and able to react in an appropriated manner to market requisites needs to have an effective support to decision-making, which means effective and flexible Data Warehousing Systems [13, 14, 15]. The application of data warehousing techniques in decision-making processes is not a new approach. Enterprises have been developing and installing specialized repositories of data oriented by subjects and competence areas, with the goal to support managers' day-by-day activities of decision-making. However, it is quite inefficient to dig in all raw data collected in the data warehouse of a project due to the fact that decisions are made to target specific enterprise's activities and area.

The requirements of an enterprise wide Decision Support System (DSS) can only be met by a system optimized for data access, quality, and integrity. The progression shown in Figure 1, illustrates the value chain of information

Data by itself is virtually useless, until it has been organized coherently in the way we can see and describe its patterns, and then we have something much more useful. These patterns are called *information*. But even seeing and describing the patterns does not lead to action. When we can identify cause and effect and correlation, then we have further refined the information into useful *knowledge*. Knowledge in this sense is not the final step. It does no good to be wise without taking action. The real, final tangible output should be decisions made as a result of the knowledge gained.

KM according to [16] is the process of generating value from an organization's intangible assets. The focus is mostly on extracting, sharing the knowledge to create value added benefits to the organization.

The main function of information technology in KM is to promote the speed of knowledge transfer [17]. Software KM tools are brought to support knowledge flows through networks and communities. There is a distinction between knowledge and information. Information is patterned data whereas knowledge incorporates set of people, principles, facts, and rules of thumb gained over experiences, and which are embedded in human action. Basically organizations need the sharing of knowledge versus simple information or data [18], [16].

Knowledge can be explicit or tacit. While *explicit* knowledge refers to the kind of knowledge that is clearly and easily formulated and documented. *Tacit* knowledge on the other hand is the type of knowledge that is generated from the subconscious [16] it is unstructured and complex, because it is generated automatically from human sensations, and has no obvious statements, which make it difficult to extract and communicate. Both types of knowledge need to be uncovered and shared across any organization.

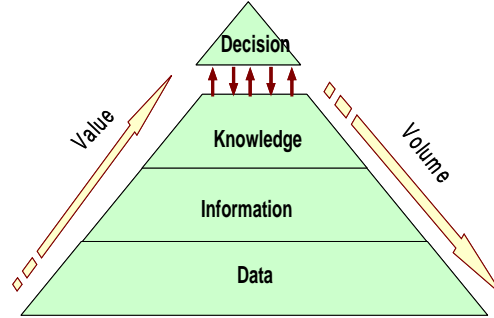


Figure1. The Value Chain of Information in a DSS System

2.2 KM at Glance

This work investigates the use of technology to expand the capacity of human's knowledge invention by taking advantages of today's high speed processing, memory and network as well as mathematical models. We deal with technology as a support for KM project. As [19] emphasizes we should build a big picture and capture a better understanding of the project phenomena, where the three main concepts: *Style*, *Adaptation*, and *Learning* should be applied at personal, project and corporate level (see Figure 2).

This work introduces a novel approach based on interacting agents that are designed to enhance the Knowledge required for the decision-making process. The intelligent agent (IA) system explores both the technology and the methods that are needed to improve the way information is gathered, managed, distributed among the users. The system characteristics include the following:

- *Intelligence*: the agent automatically customizes its self to users' requests based on previous experience. The agent also automatically adapts to changes in its environment.
- *Autonomy*: an agent is able to take the initiative and exercise a nontrivial degree of control over its own actions through the implemented roles.
- *Cooperation*: An agent does not blindly obey commands, but makes suggestions to modify requests or ask clarification questions. It also cooperates with other agents to manage the project. In order to communicate with one another, agents use language, protocol and ontology for exchanging information, knowledge, and sharing common understanding of the structure of information.

3. Literature Review and Research Methodology

A review of literature was completed to capture an understanding of how organization are applying KM to support their decision making process, and a common lesson learned is that significant number of organizations lack an accurate understanding of KM methodology. The main challenge with KM in the first place is the need of methodology that integrates it within

the overall strategic vision of the organization. The many enterprise KM projects lack a clear process, from defining the goals and requirements. Most organizations don't understand where strategically to optimize their decision-making and operational processes in an accurate manner. Another issue is that with the increasing market of KM tools, it is confusing to cross the line between those that are simply document or information management product and those that provide true knowledge management.

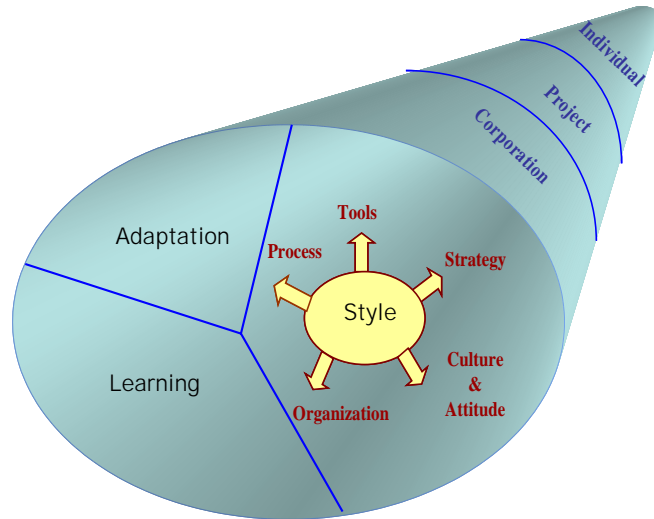


Figure 2. An integrated vision of a Project (The big picture)

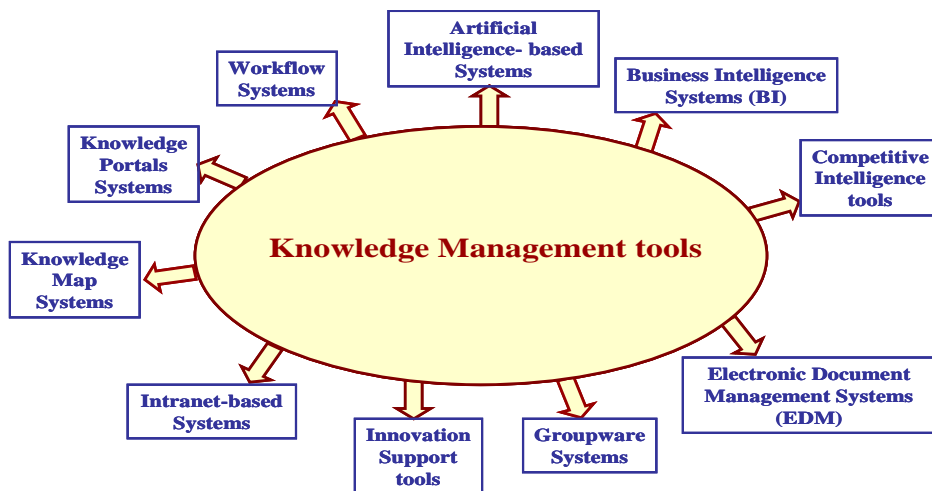


Figure 3. Software KM tool categories

It is reported that the potential of the available KM software are not well evaluated and explored. According to [18], [20] KM software is a kind of software that supports any of the

three basic KM process: generation, codification, and transfer. The results of our previous study show that KM software products can be grouped under the categories shown in Figure 3 [21].

The research method was based on gathering information through semi-structured interviews with fifteen managers, nine analysts, and twenty operational users from four organizations. This project was carried through the Fall 2003. All participants were asked to explain the available sources of data/ information/ knowledge within their organization, the extent of their usage, and how that contributes to the decision-making process.

We identified the user community based on their roles, and responsibilities for achieving the business requirement, the user communities operate within the organization and extend to partners alliance of the project, we broke the user communities into the following categories:

- *Management* refers to any individual or group of individuals that administrate or control the organization and its structures.
- *Analysts*, involve individuals who perform as their main task the assessment and diagnosis of the organization's aspects.
- *Power users*, are individuals or group who have as their auxiliary task the analysis of the aspects of the organization.
- *Individuals*, refers to operational personal.

Here are common findings and concerns:

- Intranet was found to be an important tool for finding static information.
- The implementation of electronic document management (EDM) system contributes in better retrieval, security, and version control of the enterprise document.
- They all use some sort of groupware tool such as exchange email, messenger for their geographically distributed teams.
- Data warehousing was built in all these enterprises, so information is available from their warehouse repositories. However a shortage in knowledge was reported specially in the decision that has to be made under some time constraints. Knowledge in some cases could not be found, and some times experts are not known to get their help.
- An interest for some AI techniques was raised. These techniques would be of great interest in leveraging both the reasoning and the learning process through the organization.
- An elicitation of missing to keep track of information flow was shared by the interviewees. Even with the existing scheduling, planning, and finance software tools. This was reported to be due to the lack of integration of the information held by these tools, which was limited to their users, and when communicated to the management during the regular meeting or reports it is often too late to take action.
- A strong appeal for the need of an integrated source of information from outside the organization (i.e. clients, project's partners, etc).

4. Contribution and Motivations

The idea of offering decision support always arises when timely decisions must be made in complex domains, and dynamic environments, and where multiple stakeholders are present. Support here means assistance for structuring the problem, and for analyzing and verifying the obtained structure [4], [3]. Many mission-critical decision-making happen in:

- *Dynamic* - interactive and rapidly changing information and structure.
- *Complex-cross-functional* dimension instead of traditional stand alone “single-swim-line” approach; and often
- *Unpredictable environments* - characterized by highly decentralized, up-to-date data sets coming from various sources.

A Decision Support System (DSS) designed for such situations is challenged by the need to access to the dynamically changing decentralized data at anytime from anywhere by authorized people, under tight time constraints. When taken together, this set of requirements designates the *agent technology* as the strongest solution candidate.

The choice of agents as the solution technology is motivated by the following observations:

- The domain involves an inherent distribution of data and source of knowledge both physically and logically.
- The integrity of the existing organizational structure and the autonomy of its sub-parts must be maintained.
- Interactions are fairly sophisticated, including information sharing, and coordination.

The importance of this work fits into the current efforts of companies for integrating various sources of information and knowledge to enhance their decision-making process. The proposed solution brings together benefits of the available technologies of data querying, warehousing, analysis and visualization as well as information retrieval, and knowledge sharing to its users. Those who have access will be able to make fact-based and timely decision with more degree of confidence.

5. System Architecture and Structure

An agent based integrated decision support system is proposed as a software solution with advances reasoning and learning capabilities.

As the number of agents involved in the system is distributed across organization to assist individuals, the system needs to manage itself in a scalable fashion assuring performance and extensible functionality.

The proposed system consists of a broker framework agent based. The framework provides minimal, but sufficient support for the system functionality. The complexity is kept manageable and additional functionality can be implemented in the existing agents or by adding new agents.

A snapshot of the framework extends the model of the decision support given in [5], [3] by incorporating software agent entities as shown in Figure 4. The framework is composed of:

- *Multiagents (MA)*: As we want to find something from the distributed databases, we need a multi-agents based system. At the same time, in any cases, the decision should afford on-line and dynamic information. Various agent types are used to realize the system goals. The agents' roles, capabilities, intelligence, autonomy, cooperation, communication language and protocol as well as shared ontology are considered.
- *Local Data Base Model*: this module is responsible for gathering and storing the relational database of the knowledge acquired.
- *Model Base System*: this module assists with generating alternatives (combinations of practices), and storing decision making model.
- *Knowledge based module*: for this module an expert system approach is chosen to implement the mapping of the practices. This module stores knowledge and incorporates mathematical solvers.

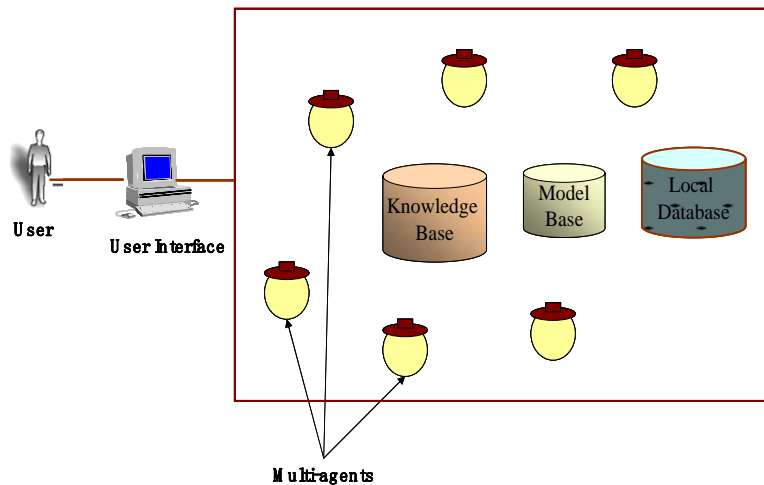


Figure 4. The DSS framework

The system is organized into four layers (Figure 5).

- *The Source layer*- This bottom layer embodies all the data sources located in an enterprise project lifecycle environment, and this includes on-line transaction processing operational (OLTP), legacy systems and data warehouse (DW).
- *The Acquisition and Integration layer*- While the previous layer guarantees the storage of

the proper data, it is up to this layer to dig inside to extract valuable information from it. Here data is extracted, transformed, and normalized towards its analytical uses. There are also all the skills concerning data navigation and their queering manipulation.

- *The Decision layer*- Decision layer of the overall system is the DSS itself. In this layer the decision making task is accepted from the user and the relative decision model is selected and the decision process is carrying out, and generates the result to the user in a more comprehensive way.
- *The Access layer*- This layer receives the user description of the decision to be performed, and deliver the results back of the decision process performed.

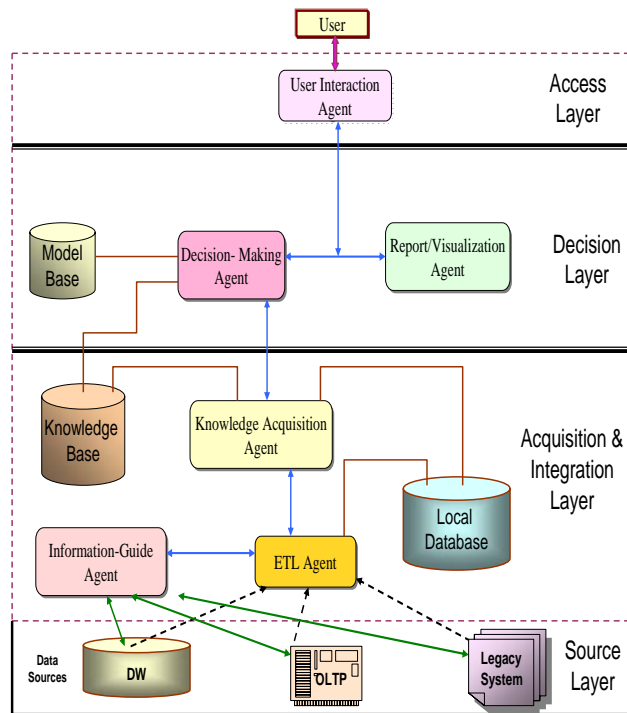


Figure 5. Overview of system architecture

5.1 System Structure

The DSS is designed using (a) *user interaction agent*, (b) *decision making agent*, (c) *report/visualization agent*, (d) *knowledge acquisition agent*, (e) *Extraction, Transformation, and Load ETL agent*, and (f) *Data-Guide agent*.

- *User Interaction Agent* interacts with the user in assisting him/her performs the decision support activities. The user can provide a general description of the problem at hand in terms of high-level goals and objectives, and decision to be performed. The user interaction agent is responsible for receiving user specifications and delivering results

- back. It also keeps track of user preferences and profile.
- *Decision-Making Agent* is responsible for coordinating the various tasks that need to be performed. From the user's specified high-level objectives, this agent converts them into specific lower level tasks and generates a plan of action. The decision-making agent has knowledge about the task domain, as well as the capabilities of other agents; it may seek the services of a group of agents that work cooperatively and synthesize the final result. Thus, the decision agent is responsible for performing the actual decision activity and providing the results.
 - *Report/Visualization Agent* incorporates the result of the decision making process and generates the final results in a more comprehensible and natural way, making it more appealing to users. This agent contains pre-formatted report templates as well as various visual representations for expressing the results of the decision support operation. It implements a variety of visualization techniques and generates the decision report.
 - The knowledge acquisition process is carried by the *Knowledge Acquisition Agent*. The acquired knowledge should be changed or modified corresponding with variety of data provided by ETL agents. Different agents should have different knowledge acquisition algorithm and feedback mechanism because they deal with different user requests. Agent should be able to valuate knowledge acquired by other agent before using it in decision-making. Furthermore, they should be able to verify or reinforce the knowledge.
 - ETL (Extraction, Transformation, and Load) process is carried by *ETL agents* who are distributed on databases that include DW, legacy and OLTP systems. These agents identify the relevant data sources, extract the data, then perform the necessary data cleansing and transforming. The extracted information then is loaded for use. An ETL agent task is the process where business data became business information.
 - *Data-guide agent* is responsible for keeping track of what data is stored where and actively maintain meta-data about each of the necessary data for the project lifecycle. It takes into account the heterogeneity of the data sources as well as resolves conflicts in data definition and representation.

5.2 Modeling of the Multi-Agents

The framework manages the distributed number of agents through the network including agents' names, life span, locations, agent creation, deletion, and migration. For the framework development both an application programming interface (API) and a run time system (RTS) are provided. The run time support is hidden in the language specific aspects of the agents and the operating system details, which implies that the code based and the operating systems are independent of the multi-agents system.

With regard to agents' interoperability, FIPA [6] standards are used to define interfaces, and protocols as well as agent implementation in terms of agent communication, and agent message transport.

To meet the manageability and scalability requirements of the system, we identify two classes of agents:

- **Application Manager Agent (AM-agent):** running on the intranet network management station.
- **Application Agents (A-agent):** refer to any type of the agents illustrated in the previous section, they run on the network and resides in a location, which is abstract from the physical location. Agent is represented by in interface, on which operation; methods can be called up on.

The difference between AM-agent and A-agent is in the authority of AM-agent, when it comes to matters of manageability of the functionality of the system such as creating destroying, moving agents.

The application manager endures policies set by human administrator, such as keeping track of agents that enter and exit the domain. AM-agent is related to the physical platform of the application domain.

The agents interaction is exclusively via message-passing, for clear and explicit organizational context and as the decision support is distributed through the organization if A-agents are part of the organization, their negotiations are more cooperative in nature than if they present A-agent project partners'. Similarly AM-agents that have agreed to hold a same project act as a team rather than a collection of individuals. Additionally, during the on going project lifecycle, more organizational grouping of agents can appear and then disband. The A-agents form themselves into collectives when they require a particular service. They combine their capabilities to offer new services that are beyond the scope of an individual A-agent. Within the integrated system, A-agents are managed using life cycle management dynamics of an agent [7].

6. System Prototype Implementation

As a proof-of concept we have implemented a prototype of the system. Since human collaboration is intrinsically peer-to-peer in nature the multiagent system is a peer-to-peer organization-wide application. The interactions are such that the agents typically interact directly with each other, without the need of an intermediary—like a server except for the directory service. Moreover, each peer typically carries its own relevant support to carry on the tasks without total need of a centralized repository.

The choice of the peer-to-peer architecture was motivated by the following:

- The peer-to-peer architecture allows information and services to no longer be gathered in a single point of accumulation. Instead, they are spread across all the nodes of the distributed system. Thus, for instance, agents (human/software) directly host the

resources they want to share with others, with no need to publish them on the server. In a sense, the server becomes now collectively represented by the whole set of available peers.

- A peer-to-peer interaction mechanism enables users to query the whole space of information provided by the currently connected peers, and find those with the relevant information or service. The uncoupling of the users' clients from the server frees clients from the coupling with servers, and at the same time allows an arbitrarily large and continuously changing set of nodes to be accessed at once [8].
- Users need to cooperate/communicate even in movement regardless where they are. Sometime an impromptu meeting needs to be set by using only their computing devices and even in the absence of a fixed network infrastructure, as allowed by wireless ad-hoc networking. Nevertheless, in this situations server access is often prevented by technical administrative barriers.
- A peer-to-peer architectural framework is more akin to scalability, reconfiguration and fault tolerance. Interestingly, these features are relevant not only in mobile scenarios but also in fixed ones, where peers can be transparently added and removed as required by the current business priorities [8]. Moreover, the decentralized nature of a peer-to-peer architecture encompasses more naturally than centralized ones, in our case, where source of information is distributed across the cooperation infrastructure.

The system was deployed using *Eclipse* technology [9]. At its core Eclipse is a platform of common tool infrastructure, including: an XML-based plug-in architecture for interoperability; frameworks for structured text and graphic editing; a common debugging infrastructure, and a complete Java development environment [10]. Furthermore, because the Eclipse core is open source, it can be easily modified, and freely downloaded. The use of Eclipse infrastructure significantly reduced the amount of effort required for the development of the system.

The agent-building platform used for this project is FIPA-OS [11]. FIPA-OS is an application domain independent agent platform, which provides agent services, i.e. message brokering and agent registration, and building blocks for implementation of the agents. According to the FIPA-OS architecture an agent application is implemented by defining application specific agent, task, conversation and message classes based on the generic ones in the platform.

The Reason for choosing FIPA-OS platform is because agents do certainly reside on multiple platforms and exhibit different behaviors. FIPA-OS can interoperate with other heterogeneous FIPA compliant platforms (i.e.: JADE) [12]. By making the agents of the system follow FIPA-OS standards we permit a large span for future enhancement of the application.

The architecture of an agent in our system is based on application specific grouping of the

agent building blocks in the FIPA-OS. The resulting overall template of each agent consists of three modules:

- *Communication module*, which provides the interface mechanisms for inter-agent communication and collaboration, this interface module is made visible to the other agents of the system and the users.
- *Operation module* which contains methods and computations that implements the process of solving a particular problem, or respond to a request from other agents or users,
- *Knowledge module* that comprehends a collection of domain-specific and domain independent knowledge appropriate to the problem solving.

Both the operation and knowledge module are private to the agent, thus, direct manipulation of the content of these modules required access privilege.

We used JESS (Java Expert System Shell) to develop the knowledge based of the agents' reasoning. JESS is an expert system shell and scripting language written entirely in Java language. It is small, light, and one of the fastest rule-based engines available. It supports the development of rule-based expert systems, which can be tightly coupled to code written in the Java language. The JESS language is very similar to the language defined by the CLIPS expert system shell, which in turn is a highly specialized form of LISP. Using JESS, we were able to build agents that have the capacity to "reason" using knowledge we supply in the form of declarative rules.

We utilized the eXtensible Markup Language (XML) for agent communication. When the peer's sender agent sends message(s) to another peer (receiving agent), this message is in a defined XML format and is transported using the Simple Object Access Protocol (SOAP). The receiver peer's agent parses the request message, processes its detail, and may return to the sender a replay also in XML format.

Agents use Oracle 9i tool to extract the data/information from the organization data warehouse. That is when user specifies his/her request that necessitates information from data source (data warehouse, OLTP, etc.) this request is converted to SQL code.

Although the implemented prototype system is still far from complete, our experience with the system indicates that the proposed DSS is a flexible and efficient tool with enormous capabilities for knowledge management towards decision-making enhancement.

7. Conclusion and Future Work

KM systems for decision support are increasingly becoming critical to organizations that wish to leverage its operational performance and improve the quality of decision making, as well as gaining critical competitive advantages. The proposed system is the first step towards this direction. In this work we opt for an intelligent agents building blocks of a peer-to-peer

system that capture the required information and knowledge for decision-making process. The multiagents proposed system brings together benefits of the available technologies of data querying, analysis and visualization as well as information retrieval, and knowledge sharing to its users. Those who have access will be able to make fact-based and timely decision with more degree of confidence. Users of the system are broadly categorized with different roles and responsibilities. Some challenges need further investigation, such as: how the agents collaborate, coordinate or compete for limited resources and how existing knowledge can be modified to match new needs.

References

- [1] P.N. Finlay, *Introducing Decision Support Systems*, Blackwell-Cambridge, MA, 274p, 1994.
- [2] S. Rochefort, "Agent-Based Transformational Architecture" International Conference on Research and Practice, Las Vegas, NV, June 2002.
- [3] E. Turban, *Decision Support and Expert Systems: Management Support Systems*, 4th Edition, Macmillan Publishing Company, NY, 1995.
- [4] G.M. Marakas, *Decision Support systems in the Twenty-first Century*, Prentice-Hall, 1999.
- [5] R.H. Sprague, H.J Watson, *Decision Support Systems -Putting Theory into Practice*, 2nd Edition, Prentice Hall, Eaglewood Cliffs, NJ, 1980.
- [6] FIPA: FIPA ACL message structure specification. <http://www.FIPA.org>.
- [7] M. Yap, N. Keong, "Are Life-Like Characteristics Useful for Autonomous Agents?" Processing of autonomous agent'99, Seattle, USA, ACM, May 1999.
- [8] G. Cugola and G. Pietro, "Peer-to-Peer for Collaboration Application", Proceeding of the 2nd International Conference ob distributed Computing Systems Workshops (ICDCW'02).
- [9] Eclipse, WWW.eclipse.org
- [10] D. Rayside and M. Itoiu, "Visualizing Flow Diagrams in WebSphere Studio Using SHriMP Views", Kluwer 2003.
- [11] Fipa-Os, <http://fipa-os.sourceforge.net/>
- [12] T. Peddireddy, and J. Vidal, "Multiagent Network Security System using FIPA-OS". Proceedings IEEE SoutheastCon2002
- [13] W. Inmon, "Building the Data Warehouse," Wiley, 1996.
- [14] P. Gray and H. Watson, *Decision Support in the Data Warehouse*, the Data Warehousing Institute Series from Prentice Hall, 1998.
- [15] R. Barquin and H. Edelstein, editors, "Building, Using, and Managing the Data Warehouse," The Data Warehousing institute, Prentice Hall, 1997.
- [16] J. Leibowitz, "Knowledge Management and Its Link to Artificial Intelligence," Elsevier 2001.
- [17] T. Davenport and L. Prusak, "Working Knowledge: How Organizations Manage What They Know," Boston: HBS Press, 1998.

- [18] C. Rodrigo, B. De Mata and A.T. Ferreira, "Using information to support knowledge conversion process." Information Research, 2001.
- [19] A.J. Shenhar, "Strategic project management: the new Framework", IEEE 1999. Proceedings.
- [20] Nonaka and H. Takeuchi, "The Knowledge-Creating Company." New York: Oxford Press, 1995.
- [21] N. Houari, B.H. Far, "Application of Intelligent Agent Technology for Knowledge Management Integration", Proc.3rd (Ed) IEEE International Conf. Cognitive Informatics IEEE CS Press, 2004, pp.271-279.