Multi-Agent System for Knowledge-Based Access to Distributed Databases

Priti Srinivas Sajja
Sardar Patel University, Gujarat, India

priti_sajja@yahoo.com

Abstract

A framework for knowledge discovery, knowledge use, and knowledge management is presented in this article to provide knowledge-based access of the domain databases using multi-agent systems approach. This framework encompasses five different agents: namely, knowledge management agent, data filter agent, rule induction agent, dynamic analysis agent, and interface agent. This article suggests an enhancement in the typical Knowledge Query and Manipulation Language (KQML) used to interact recurrently and to share information between multiple agents to achieve their goals by including the notion of linguistic variable and, hence, to support fuzzy decision making. The article also includes a sample KQML query block (along with membership function used by the knowledge management agent), result of the query, and structure of database files for a co-operative dairy. The approach provides advantages like effectiveness, explanation, reasoning, multimedia, and user-friendly interface in accessing multiple databases for an application.

Keywords: knowledge-based systems, multi-agent system, knowledge query and manipulation language, linguistic variable

Introduction

Knowledge-Based Systems (KBS) are productive tools of Artificial Intelligence (AI) working in a narrow domain to impart quality, effectiveness, and knowledge-oriented approach in decision-making process. Being a product of fifth generation computer technology, KBS possess characteristics like (Efraim, 1993):

- providing a high intelligence level;

- assisting people to discover and develop unknown fields;

- offering vast knowledge base;

- aiding management activities;

- solving social problems in better way;

- acquiring new perceptions by simulating unknown situations;

- offering significant software productivity improvement; and

- reducing cost and time to develop computerized systems.

One of the main components of KBS is the knowledge base, in which domain knowledge, knowledge about knowledge, factual data, procedural rules, business heuristics, and so on are available. The inference engine is another component, which infers new knowledge and utilizes existing knowledge for decision-making and problem solving. Explanation/reasoning and self-learning are two more components to improve acceptability and scope of the system. These components also provide justification for the decision taken. Additionally, a user interface is available to interact with users in more friendly way. Figure 1 shows position of the KBS in the well-known data pyramid along with its general structure.
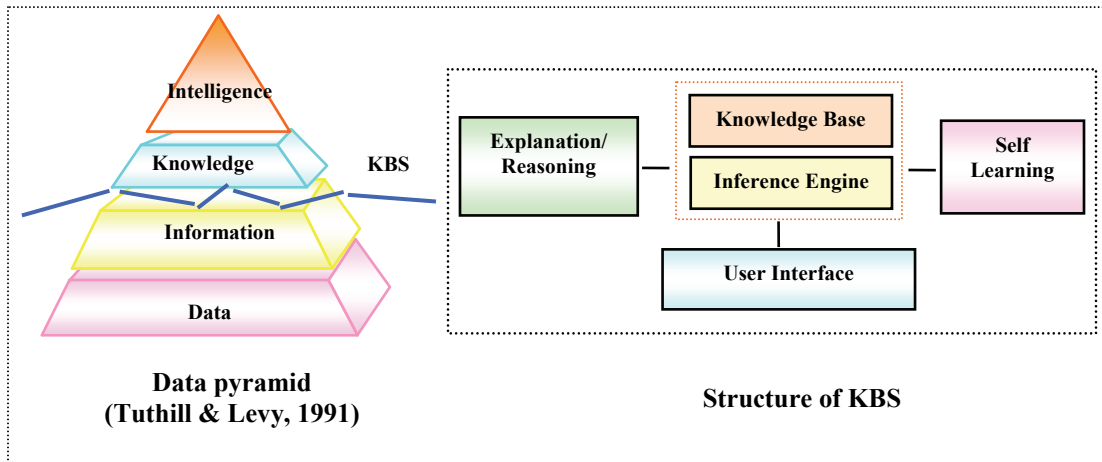

Figure 1: Data pyramid and KBS

Typical relational database management systems deal with data stored in predefined format in one or more databases/tables. These systems do not deal with knowledge and/or decision processing and do not include features like:

- capability to add powers to the solution and concentrate on effectiveness;
- transfer of expertise, use of expertise in decision making, self learning, and explanation;
- mainly symbolic manipulation;
- learning by case/mistakes;
- ability to deal with partial and uncertain information; and
- work for narrow domain in a proactive manner.

In the information and communication technology era today, a large number of processes is automated and generates number of large databases. Some applications span their boundaries in multiple dimensions and deal with multiple databases in a distributed fashion. Such large databases in business contain staggering amounts of raw data. These data must be looked at to find new relationships, emerging lines of the business, and ways for improving it. Trying to make sense out of these data requires a knowledge-oriented perspective, which is not easily achieved through either statistical process or even multidimensional visualization alone (Cox, 2005). The potential validity or usefulness of data elements or patterns of data elements may be different for various users. The relevance of such items is highly contextual, personal, and changing continuously. According to Donovan (2003), making retrieved data or a description of data patterns generally understandable is also highly problematic. Moreover, the structure and size of the data set or database and the nature of the data itself make the procedure more complex and tedious. This leads to the need

for the proposed system in which databases can be accessed in knowledge-oriented fashion. To achieve this, productive agents like KBS can be utilized to search and manage database content to impart quality and effectiveness. Section two of this paper proposes a framework and methodology of knowledge-based access to multiple databases using modified Knowledge Query and Manipulation Language (KQML) as communication means between agents. Section three discusses an illustrative situation along with the structure of databases, a sample agent communication using KQML block, and a typical query by an agent to another agent with an example in dairy industry that works on the proposed architecture.

Multi-agent System Architecture

The term 'agent' is loosely defined. However, an agent can be referred to as a component of software and/or hardware, which is capable of acting exactly like a user in order to accomplish tasks on behalf of its user. KBS tools used as agents are autonomous, co-operative, and able to learn themselves. A multi-agent system can be considered as a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (Durfee, Lesser, & Corkill, 1989; Sajja, 2005). A multi-agent system comprised of multiple autonomous components needs to have certain characteristics (Jennings, Sycara, & Wooldridge, 1998; Roberto, 1999):

- each agent has incomplete capabilities to solve a problem;

- there is no global system control;

- data is decentralized; and

- computation is asynchronous.

That is, combining multiple agents in a framework presents a useful software engineering paradigm where problem-solving components are described as individual agents pursuing high-level goals.

As most business applications deal with several databases of a homogeneous nature, they can interact easily. However, such interaction and content retrieval is limited in its scope and is static. In addition to this, to access the databases in knowledge-based fashion, explicit (manual) expertise becomes essential. Such expertise includes tasks like meta-knowledge and domain knowledge management, filtering and statistical analysis of data from the databases, interface, and presentation related tasks. Most of these tasks have their own methodology and are highly independent from the other tasks though carried out for common system objectives. Moreover, for the standard tasks like interface, data analysis, and data retrieval, mechanisms once developed can be reused for other systems to increase reusability of the system. This leads to development of necessary components as different agents for every specific independent task within a common framework enabling these agents to interact. The multi-agent systems developed so-far are application specific and can not be reused. Gibert et al (2002) developed a system using statistical and knowledge management agents, specifically for management of environmental databases for effective decision support systems. Another example is an agent-based intelligent environmental monitoring system by Ioannis & Pericles (2004), which presents a multi-agent system for monitoring and assessing air-quality attributes and which uses data coming from a meteorological station.

There is a need for multi-agent systems that can manage databases of commercial applications in more generalized and implicit way. The architecture presented in this section encompasses multiple agents in a collaborative fashion and works as a prototype for knowledge-based data analysis and implicit knowledge management of one or more databases. Figure 2 describes the generalized frame work in three different phases, namely knowledge discovery, knowledge management, and knowledge use. Discovered knowledge is presented to the users with the help of user interface.
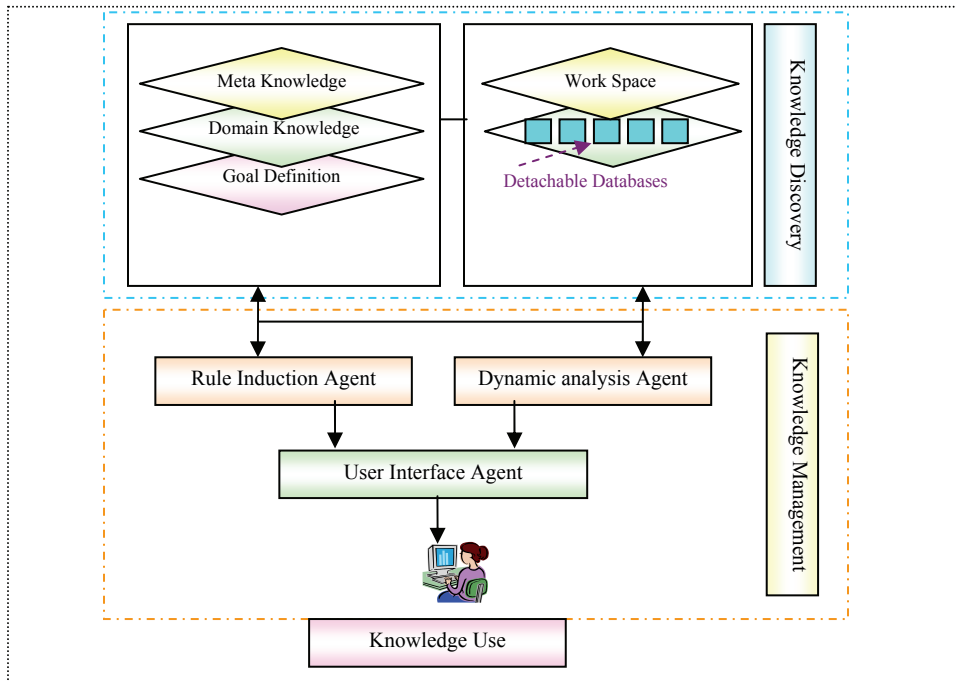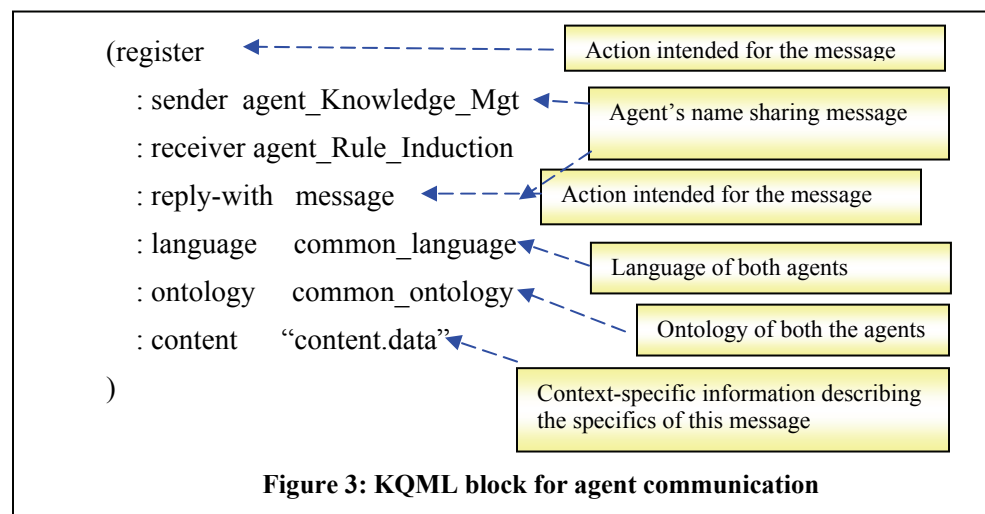
Figure 2: Architecture of multi-agent system

The broad objectives of different agents in architecture shown in Figure 2 are as follows:

Name of the Agent	Purpose
Knowledge management agent	This agent takes care of management processes like storage, retrieval and inference of knowledge chunks according to the goal definition. The agent contains (i) knowledge about knowledge, (ii) knowledge about processes, (iii) knowledge about users, and (iv) knowledge about goal definition. This component is responsible for activities like integration, validation, and utilization of different knowledge patterns acquired.
Rule induction agent	This agent works on decision tree induction, classifies, and forwards the rules to other agents.
Data filtering agent	The prime objective of this agent is to filter data from the domain files, set parameters and suggestions of method under the assistance of the knowledge management agent and rule induction agent. Such a database agent can also keep track of the drivers used by each database and manage the details like data formats, access details, and storage mechanisms. According to Srivastava et al (2005), this allows agents to query every type of database and makes systems flexible.
Dynamic analysis agent	This agent uses dynamic analysis and statistical analysis methods for analyzing filtered data under the assistance of the knowledge management agent.
User interface agent	This agent provides a multimedia user interface to end-users.

All these agents work in direct/indirect assistance of the knowledge management agent. The agents in a multi-agent system should recurrently interact to share information and to perform tasks to achieve their goals. Researchers investigating agent communication languages mention three key elements to achieve multi-agent interaction (Finin, Labrou, & Mayfield, 1997; Huhns & Singh, 1998; Peng et al, 1998):

- a common agent communication language and protocol;
- a common format for the content of communication; and
- a shared ontology.

The agents can interact with a declarative communication language like KQML (Genesereth & Fikes, 1992). It is conceived both as a message format and a message handling protocol to support run-time knowledge sharing among agents (Finin et al., 1997). An example of KQML block is shown in Figure 3.

```
(register                          ← ——————————  Action intended for the message
    : sender  agent_Knowledge_Mgt  ←——  Agent's name sharing message
    : receiver agent_Rule_Induction
    : reply-with   message         ←———  Action intended for the message
    : language    common_language  ←  Language of both agents
    : ontology    common_ontology  ←  Ontology of both the agents
    : content     "content.data"   ←  Context-specific information describing
)                                      the specifics of this message
```

Figure 3: KQML block for agent communication

These KQML agents may also be considered intelligent, though not often mobile. Genesereth and Ketchpel (1994) proposed an excellent description of this kind of software agent. User defined predicates can be developed using KQML syntax as needed. Few examples are as follows (Scott, 1999):

Example	Meaning
bel(A, B)	A believes in B
Complete(A)	A is complete
do(A, B)	A does task of B
evaluate(A)	Evaluate expression of A
forward(A, B)	A forwards message to B
isAbleTo(A, B)	A is able to do task of B

The KQML can be enhanced with fuzzy linguistic variables in its tags/predicates to deal with the human style of decision processing. Fuzzy logic was formulated by Zadeh (1965) as multi-valued logic between 0 and 1. Fuzzy logic is widely applicable especially in system controls that are very complex, uncertain, and cannot be modeled precisely even under various assumptions and approximations. Values of such fuzzy linguistic variables are not crisp, but fuzzy, such as low…medium…high…poor… moderate…rich. This offers a framework for approximate reasoning and allows qualitative knowledge about a problem to be implemented into the systems using

fuzzy rules. This makes the system more user friendly, readable, and easy to maintain and effectively reduces the complexity by reducing the number of rules.

Using this concept, a multi-agent system that consists of business databases and different agents along with predicates developed for knowledge-based access can be developed. The system should also provide means to access the domain databases, means for providing key words for interaction, and space to view the result. An illustration for the same is described in the following section.

An Illustration of Different Elements of the System

To use the data stored in large databases generated through business processes for effective decision making, the above configuration can be utilized. As an illustration, the database of a co-operative dairy is shown here. Figure 4 shows the working database files (about co-operative societies, milk procurement, and member information) along with a sample communication between agents (sender agent: agent_dynamic_analysis and receiver agent: agent_user_interface) and the resulting graph. The control panel buttons (such as previous, next, forward, and help) facilitate traversing through database records for a selected file. The databases generated by the business procedures are about (i) users, employees, co-operative society information, wholesalers, retailers, etc; (ii) facilities and major processes; and (iii) outputs, results, and history. The data filtering agent considers these database files to filter business data and provide filtered results in the workspace. The interface agent provides access to the workspace through the data filter agent, enables users to write a query in common ontology (modified KQML here), and provides visualization of result.

A sample KQML block is given below:

```
(register

        : sender        agent_dynamic_analysis

        : receiver      agent_user_interface

        : reply-with  message

        : language    common_language

        : ontology    common_ontology

   : content      "content.data" )
```

Here content.data in the above KQML block describes the multi liner query message. Using this facility, one may write the query in a user friendly way as follows:

Open member information database

Select all members with high milk production per month

Draw chart with option ('3d_Graph', x:co-operative society code, y:avg production, scale:default)

The fuzzy membership definition function (*high production*) used for the sample content file (*content.data*) is shown in Figure 5.
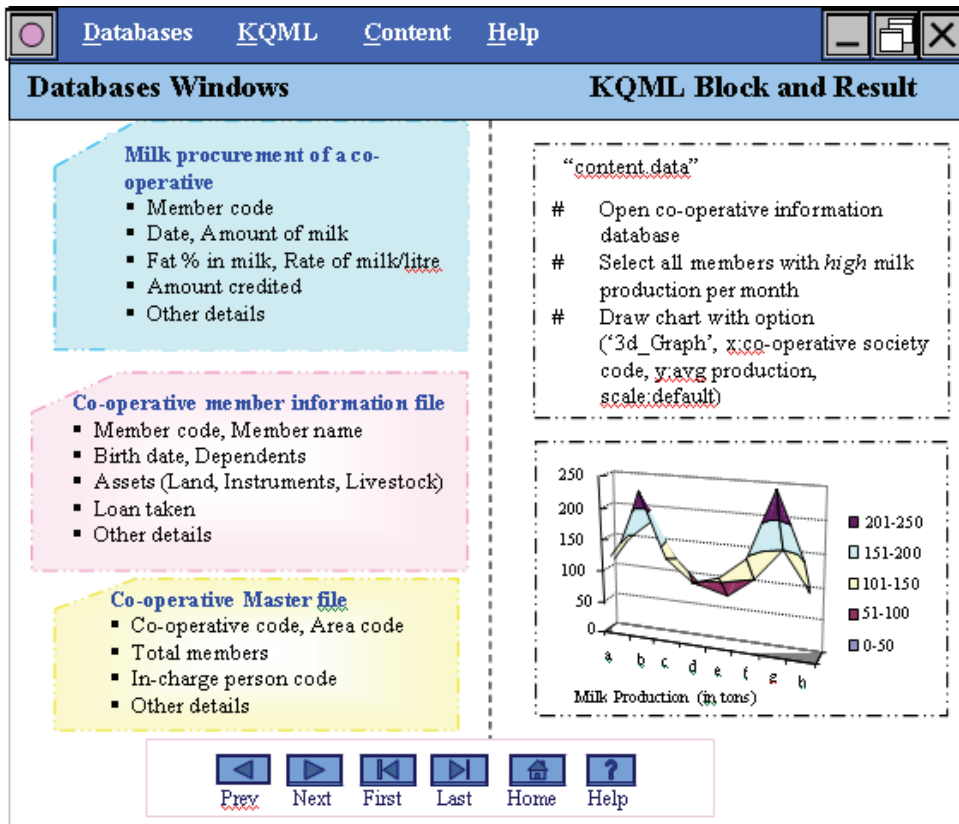
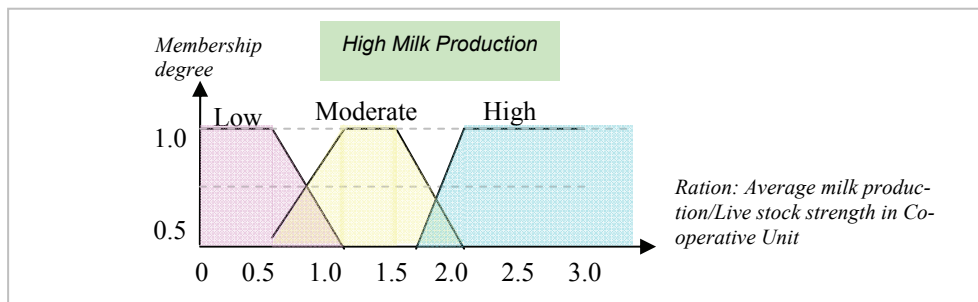**Figure 4: Elements of screen design for the multi-agent system**

Figure 5: Linguistic variable 'high' production

Conclusion

Knowledge management, discovery, and use of knowledge are the three different phases supported by the agents of the proposed structure. Various applications that use databases can be considered as candidates for the framework for knowledge-oriented access of the content to provide advantages like effectiveness, explanation, reasoning, multimedia, and user friendly interface in more structured way.

It is assumed that all databases of the concerned application(s) need to share information and data with each other and, hence, they are following one fix modeling topology like relational database management systems (RDBMS). Observing this, the multi-agent system presented here manages the existing databases in a knowledge-based fashion, provided they are following one fixed mod-

eling topology; here it is RDBMS. The structure does not support databases with different representation methodologies. The work can be extended by considering heterogeneous databases within the same framework, if required.

Corporate sector, academic sector, R & D, and industries such as manufacturing, banking, medicine, entertainment, and service generate large amounts of data, which are available in the form of database files. In this scenario such implicit and automatic knowledge management of the available databases incorporated in the given application has strong potentials. The knowledge-based access of the domain information can enhance the utilization of scarce resources for development and, hence, lead to synergic development.

References

Cox, E. (2005). *Fuzzy modeling and genetic algorithms for data mining and exploration.* CA: Morgan Kaufmann Publications.

Donovan, A. M. (2003). Knowledge discovery in databases and information retrieval in knowledge management system. Retrieved January 6, 2007, from www.ischool.utexas.edu/~i385tkms/blog/archives/Donovan_385Tpaper_revision.doc

Durfee, E. H., Lesser, V. R., & Corkill, D.D. (1989). Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering, 1*(1), 63-83.

Efraim, T. (1993). *Decision support and expert systems* (3rd ed.). NY: Macmillan Publishing Company.

Finin, T., Labrou, Y., & Mayfield, J. (1997). KQML as an agent communication language. In J. M. Bradshaw (Ed.), *Software agents* (pp. 291-316). CA: AAAI Press.

Genesereth, M., & Fikes, R. (1992). *Knowledge interchange format.* Version 3.0 Reference Manual, Technical Report Logic 92-1, Computer Science Department, Stanford University.

Genesereth, M., & Ketchpel, S. P. (1994). Software agents. *Communication of the ACM, 37*(7), 48-53.

Gibert, K., Sànchez-Marrè, M., Rodríguez-Roda, I., Bueno, E., Mozo, L., Clavell, A., Martín, M. & Rougé, P. (2002). Development of an intelligent data analysis system for knowledge management in environmental databases. *Proceedings of the First Biennial Meeting of The International Environmental Modeling and Software Society (iEMSs 2002),* Manno, Switzerland, 420-425.

Huhns, M. N., & Singh, M. P. (1998). Agents and multi-agent systems: Themes, approaches, and challenges. In M. N. Huhns & M. P. Singh (Eds.), *Readings in agents* (pp.1-23). San Francisco: Morgan Kaufmann Publishers.

Ioannis, N. A., & Pericles, A. M. (2004). An agent-based intelligent environmental monitoring system, *Management of Environmental Quality, 15*(3), 238-249.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems Journal, 1*(1), 7-38.

Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W. J. & Boughannam, A. (1998). A multi-agent system for enterprise integration. *Proceedings of the Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 155-169.

Roberto, A. F. (1999). Towards standardization of multi-agents systems framework. *Crossroads, 5*(4), 18-24.

Sajja, P. S. (2005). Multi-layer knowledge-based system: A multi-agent approach. *Proceedings of the Second Indian International Conference on Artificial Intelligence*, Pune, India, 2899-2909.

Scott, A. M. (1999). KQML & FLBC: Contrasting agent communication languages. *Proceedings of the 32nd Hawaii International Conference on System Sciences, Maui, Hawaii*, USA, 29-32.

Srivastava, A., Giri, N., Nirav, G., Devraj, B., Tushar, B., & Manas, K. (2005). Virtual enterprise over COBMAF: A CORBA based multi-agent framework. *Proceedings of the Second Indian International Conference on Artificial Intelligence*, Pune, India, 2938-2955.

Tuthill, G .S., & Levy, S. T. (1991). *Knowledge-based systems: A manager's perspective*. TAB Professional and Reference Books.

Zadeh, A. L. (1965). Fuzzy sets. *Journal of Information and Control, 8*(3), 338-353.

Biography

Dr **Priti Srinivas Sajja** (b. 1970) joined the faculty of the Department of Computer Science, Sardar Patel University, India in 1994 and presently working as a Reader. She received her M.S. (1993) and Ph.D (2000) in Computer Science from Sardar Patel University.

Her research interests are predominantly in the area of artificial intelligence. She has 31 publications in journals and in the proceedings of national and international conferences. Two of her publications have won best research paper awards. She is member in editorial board of four international science journals based at USA and Brazil.