A MULTI-AGENT MODEL AND ARCHITECTURE FOR MULTI-MODAL KNOWLEDGE ASSISTANCE AND CAPITALIZATION

ZAFAR IQBAL HASHMI

UNIVERSITI SAINS MALAYSIA
2004

A MULTI-AGENT MODEL AND ARCHITECTURE FOR MULTI-MODAL KNOWLEDGE ASSISTANCE AND CAPITALIZATION

by

ZAFAR IQBAL HASHMI

Thesis submitted in fulfillment of the requirements
for the degree of Masters of Science

JUNE 2004

ACKNOWLEDGEMENT

I would like to gratefully acknowledge the supervision of Assoc. Prof. Dr. Syed Sibte Raza Abidi for many insightful conversations during the development of the ideas in this thesis, for helpful comments, for his dedication, guidance, and for having presented me with the numerous opportunities to broaden my academic and social horizons.

I am indebted to Dr. Bahari Belaton and Prof. Dr Zaharin Yusoff for their guidance and assistance during my work. I would like to thank heartedly Dr. Cheah Yu-N for his kind assistance and suggestions as one of the readers of my thesis, as a friend, as a teacher and for his impressive way of conveying thoughts.

I am thankful to my close friends for the enjoyable moments we shared over the past few years. I would like to acknowledge the thanks to my colleagues at the Health Informatics Research Group: Syed Zahid Hassan, Selvakumar Manickam, Chong Yong Han, Seelan, Shailendra Singh, and Leong Wan Sun for their support and fellowship.

I am extremely grateful to my parents: Muhammad Iqbal Hashmi and Suriya Iqbal, for their love, care, nurturing, their perseverance and determination for encouraging me always to accept challenges and to pursue a post-graduate degree. I thank my brothers and sisters: Azhar Iqbal Hashmi, Faisal Iqbal Hashmi, Saima Iqbal Hashmi, and Asma Iqbal Hashmi for their love and moral support. My special thanks to Somaia for her affection, encouragement, moral support, her lovely nature, for pushing me to complete my thesis and for always being there for me. Finally, I am thankful to Allah for his blessings and love as the eternal guardian for me through all my life. All powers and glory goes to Him, now and forever.

TABLE OF CONTENTS

Chapter 8

LISTS OF TABLES AND FIGURES

ABSTRAK

MODEL DAN SENIBINA BERBILANG AGEN UNTUK BANTUAN DAN PERMODALAN PENGETAHUAN BERBILANG MOD

Perubahan paradigma yang disebabkan oleh perkomputeran berasaskan agen telah membolehkan penghasilan peralatan, teknik, strategi dan methafora yang berkuasa yang membolehkan pembangunan dan pengkonsepsian pelbagai perisian. Masyarakat perisian telah mula untuk mempercayai bahawa perkomputeran berasaskan agen akan membawa keputusan yang memberangsangkan ke dalam arus pembangunan utama sains komputer. Walaupun sistem agen cerdas dan berbilang agen mempunyai banyak potensi yang menguntungkan dan mempunyai mekanisma yang unggul, tetapi masih menghadapi pelbagai cabaran daripada pelbagai aspek berlainan seperti permodelan, analisis, rekabentuk, interaksi, koordinasi, kerjasama, senibina dan lain-lain lagi. Isu-isu ini telah membantutkan adaptasi teknologi ke dalam arus pembangunan utama sains komputer.

Dalam tesis ini kami telah membangunkan sebuah metodologi berbilang langkah, 3 fasa yang diikhtiarkan untuk mengatasi masalah berkaitan dengan sistem agen cerdas dan berbilang agen dan membangunkan sebuah sistem aplikasi berbilang agen yang berasaskan teknik, strategi dan mekasnisma dalam tesis ini.

Dalam fasa pertama penyelidikan, pertama sekali, kami telah membangunkan model Agen Berautonomi Cerdas (ABC) Hibrid yang menyediakan senibina dari segi teori dalam mendefinasikan spesifikasi untuk rekabentuk agen berautonomi cerdas tunggal. Keduanya, kami telah mendefinasikan sebuah rekabentuk generik Hibrid ABC berasaskan kepada model Hibrid ABC. Rekabentuk generik ABC kami boleh bertindak secara perundingan dan reaktif untuk meniru kelakuan seperti manusia. Ketiga, kami telah membangunkan Model Sistem Berbilang Agen jenis Kerjasama yang dipanggil Kerjasana Berasaskan Kehendak (KBK). Model ini akan menentukan skema kordinasi dan kerjasama dengan protokol komunikasi yang bersesuaian untuk mengurangkan komunikasi dan penggunaan sumber yang berlebihan. Keempat, kami telah membangunkan bahasa komunikasi agen yang dipanggil bahasa perihalaan pengetahuan (BPP) yang berpegang kepada protokol yang dibincangkan sebelumnya.

Dalam fasa kedua penyelidikan, pertama sekali, kami telah membangunkan sebuah metodologi yang berasaskan agen yang cerdas dan berautonomi untuk analisis dan rekabentuk yang berasaskan kepada pandangan teori agen dan mengambil kira sturuktur organisasi, model organisasi, protokol kerjasama, perwakilan interaksi luaran dan rekabentuk sistem multi-agen. Kedua, kami telah membangunkan sebuah senibina Sistem Berbilang Agen Cerdas bagi Bantuan dan Permodalan Pengetahuan Perbubatan (BACBPPP) yang berasaskan Model Sistem Berbilang Agen jenis Kerjasama Berasaskan Kehendak.

Dalam fasa ketiga dan terakhir dalam penyelidikan telah membangunkan sistem (BACBPPP) untuk sistem berbilang agen yang membawa kesemua teknik, metodologi dan strategi yang telah didefinasikan dalam tesis bersama.

Kami telah menjalankan satu ujian kuantitatif dan kualitatif ke atas kerja yang dilaporkan dalam tesis ini. Ujian qualitatif telah dijalankan untuk Model ABC Hibrid, senibina generik ABC Hibrid, model berbilang agen KBK, metodologi berasaskan Agen Berautonomi Cerdas dan KDL. Sementara itu ujian kuantitatif telah dijalankan untuk system berbilang agen BACBPPP. Keputusan ujian untuk model ABC Hibrid menunjukkan bahawa model yang dicadangkan lebih realistik, cerdas, menyerupai tingkah laku manusia dan memberikan lebih kawalan kognitif dan fleksibel daripada model BDI. Rekabentuk generik ABC Hibird telah dibandingkan dengan senibina PRS dan DECAF dan keputusan menunjukkan bahawa ABC Hibrid mempunyai ciri-ciri tambahan yang menjadikannya pilihan yang lebih baik untuk mengatasi masalah yang terdapat dalam PRS dan DECAF dari segi penjadualan, piawaian definisi mesej, mekanisma komunikasi yang fleksibel, meniru tingkah laku manusia, merancang dan rancang semula, masa tindakbalas yang pendek dan permulaan yang mudah. Model berbilang agen KBK menunjukkan keputusan positif dalam mengurangkan overhed komunikasi di antara berbilang agen terhadap model berbilang agen yang lain. Penilaian ke atas KDL menunjukkan yang ianya lebih baik dari segi mendesripksi pengetahuan, senang untuk diterangkan, kerumitan pentaakulan dan perwakilan berasaskan matlamat berbanding bahasa komunikasi lain seperti KIF dan LARKS. Keputusan penilaian untuk metodologi berasaskan Agen Berautonomi ABC menunjukkan bagaimana ia berjaya mengatasi masalah dan memberikan strategi yang lebih baik untuk penganalisaan dan rekabentuk suatu agen berbanding GAIA dari segi struktur organisasi, senibina berbilang agen, model organisasi, rancangan dan ketulenan orientasi agennya. Kami telah melakukan suatu penilaian kuantitatif ke atas sistem berbilang agen BACBPPP yang menunjukkan tahap kebolehpercayaan yang tinggi sistem ini dan nilai ingat dan kejituan yang tinggi. Kami dapati tahap kebolehpercayaan dan keefisian menjadi tinggi disebabkan pembahagian index pengetahuan, pembahagian ontologi, sifatnya yang kerotonomi atau serentak, dan lokasi agen yang teragih.

Kesimpulannya, penyelidikan kami dalam permodelan,rekabentuk dan pembangunan sesuatu yang menarik yang menuju ke arah penghasilan idea-idea yang baru dan teknik-teknik yang ber inovasi bagi kemajuan teknologi agen.

ABSTRACT

The paradigm shift set by agent-based computing provides powerful tools, techniques, strategies and metaphors that have the potential to significantly improve the way in which people conceptualize and develop different kinds of software. The software community has started to believe that agent-based computing is the next significant breakthrough for the mainstream of software development. Although intelligent agents and multi-agent systems provide many potential advantages and state of the art mechanisms to solve problems, they face many difficult challenges from different aspects like modeling, analysis, designing, interaction, coordination, cooperation, architectures etc. These issues hinder the adoption of technology into the mainstream of computer science.

In this thesis we have developed a contrived, multi-step, three phase methodology to address some of the problem issues related to intelligent agents and multi-agent systems and developed a multi-

agent system application based on the techniques, strategies and mechanisms developed through out this thesis.

In the first phase of our research, firstly, we developed *Hybrid Intelligent Autonomous Agent (IAA) model* that provides the theoretical framework in defining the specifications for single intelligent autonomous agent architecture. Secondly, we defined, a *Hybrid Intelligent Autonomous Agent (IAA) generic architecture* based on the hybrid IAA model. Our IAA generic agent architecture can act in a deliberative and reactive manner and it incorporates motivators to mimic human-like behavior. Thirdly, we developed *Demand Based Cooperation (DBC) Cooperative Multi-agent System Model*. The DBC multi-agent system model is a cooperative multi-agent model that specifies the coordination and cooperation scheme with appropriate communication protocols to lessen the communication and system resources overheads. Fourthly, we developed a content level agent communication language called *Knowledge Description Lanaguage (KDL)* that adheres to the protocols defined in the DBC multi-agent model.

In the second phase of our research firslty, we developed an *Intelligent Autonomous Agent (IAA)-oriented Methodology* for analysis and designing of agents which is based on the holistic view of agent theory and takes into account the organizational structure, organizational model, cooperative protocols, explicit interaction representation and designing of the multi-agent system architecture. Secondly, we developed an *Intelligent Healthcare Knowledge Assistance and Capitalization (IHA) Multi-agent system architecture* that is based on the DBC cooperative multi-agent model.

In the third and final phase of our research, we developed the IHA system for multi-agent system which brings together all the techniques, methods and strategies defined in this thesis.

We conducted a qualitative and quantitative evaluation of the work presented in this thesis. Qualitative evaluation was conducted for Hybrid IAA model, Hybrid IAA generic architecture, DBC multi-agent model, Intelligent Autonomous Agent (IAA)-oriented methodology, and KDL while quantitative evaluation was conducted for IHA multi-agent system. Evaluation results for Hybrid IAA model showed that the proposed model is more realistic, intelligent, mimics human behavior and provides more cognitive and flexible control than the BDI model. The Hybrid IAA generic architecture was evaluated against PRS and DECAF architectures and results showed that the Hybrid IAA has a number of extra characteristics which makes it a better choice by overcoming problems found in PRS and DECAF in terms of better scheduling, message definition standards, flexible communication mechanisms, mimicking human behavior, planning and re-planning, short response time and simple initialization. The DBC multi-agent model showed positive results in the reduction of communication overhead among multi-agents against other multi-agent models. Evaluation of KDL showed that it is better in terms of richness of knowledge description, extensibility, ease of expressiveness, reasoning complexity and goal-oriented representation over other agent communication languages such as KIF and LARKS. The results of the evaluation for the IAA-oriented methodology showed how it overcame the problems and provided a better strategy for the analysis and design of agent compared to GAIA in terms of organizational structure, multi-agent architecture, organization model, plans and pureness of its agent orientation. We conducted a quantitative evaluation of the IHA multi-agent system which showed high reliability of the system, and high *Recall* and *Precision* values. We found that reliability and efficiency became high because of the division of knowledge index, division of ontology, its autonomous/concurrent nature, and the distributed locations of agents.

In conclusion, our research on the modeling, designing and development of the intelligent agents and the multi-agent systems designing, modeling and development is an exciting one, leading towards generation of new and innovative techniques for the advancement of agent technology.

Chapter 1

INTRODUCTION

> *"Agents are here to stay, not least because of their diversity, their wide range of applicability and the broad spectrum of companies investing in them. As we move further and further into the information age, any information-based organization which does not invest in agent technology may be committing commercial hara-kiri."*
>
> *—Hyacinth S. Nwana*

1.1 Introduction

Agent software is a rapidly developing area of research. The word agent is currently in vogue in the popular computing press as it is within the artificial intelligence (AI) and computer science communities (Chauhan, 1997). Agent-based computing provides an exciting new paradigm and synthesis both for artificial intelligence and, more generally, computer science problem solving, and has the potential to significantly improve the theory and practice of modeling, designing and implementing computer systems (Nicholas, 2000). Intelligent agents are an emerging technology that is making computer systems easier to use by allowing people to delegate work back to the computer. They help do things like find and filter information, customize views of information, and automate work. "In our lives, when we find ourselves in a position where time and activities overtake us, we seek help in the form of assistants - people who take care of things we could do ourselves, but prefer not to. In the computer world, intelligent agents play the role of such assistants" (Gilbert, 1997). One important aspect of the current concept of agent is that an agent acts on behalf of its human user to perform one or more portions of the overall task of gathering and using information. Unlike an ordinary program on a single computer, an agent may run on any location or site and report its results to any other location or site or to another agent via the network. This is contrasted to the traditional programming paradigm where all of the resources and their locations are known prior to execution of the program or very shortly after execution begins. A key element in the agent programming paradigm is that of adaptation. Adaptation is essential in multi-agent systems since the availability of resources may change, agents may not be able to predict results of information requests and environment changes may require task recognition (Graham, 2001). Agent applications are being developed for fields as varied as manufacturing, entertainment, electronic commerce, user assistance, service and business management, information retrieval, research, etc. According to BIS Strategic Decisions, "Agents will be the most important computing paradigm in the next 10 years".

1.2 Motivation Towards Intelligent Agent Technology

Agent based computing has been considered the new revolution in software. One of the interesting and important aspects of agents is that they facilitate cognitive modeling (based on behavioral aspect fulfilling a particular purpose), as opposed to role modeling (based solely on purpose) (Rahwan *et al*., 2000). Agents are different from current programs in three ways: 1) they are proactive, i.e. they initiate actions, 2) they are adaptive, i.e. they learn from the user's preference and habits, and 3) they are personalized, i.e. they change their way of helping the user according to what they have learned about the user (Chavez *et al*., 1997). Software agents work autonomously without waiting for a user's command. The user and the agent both communicate to each other to complete a task. As we depend more on computers to complete our daily tasks, we will have to rely for some part of our productivity on software agents, or "digital assistants". As Maes (1994) says, "we evaluate such digital assistants using two criteria: their ability to perform a task on behalf of the user, and their trustworthiness, or the assurance that allows the user to delegate tasks to an agent without worrying. We will only adopt agents if we can trust them and control their actions".

Agents assist users in various ways and domains. The tasks suited for computer agents are the same as those to which we employ human agents, i.e. tasks that require expertise, skill, resources, or labor to accomplish some goal (Maes, 1994). We can categorize those tasks as follows:

- Information-based tasks: navigation and browsing, information retrieval, sorting, organizing, and filtering.
- Work-related tasks: reminding, programming, scheduling, advising, and managing emails. Learning: coaching, tutoring, and assistance.
- Entertainment: playing against, playing with, and performing.

The agent enables a person to state what information he or she requires and the agent then determines where to find the information and how to retrieve it (Etzioni and Weld, 1995). Autonomous, intelligent agents may prove to be the key requirement in transforming passive search and retrieval engines into active, personal assistants (Jansen, 1996).

1.3 Agent Models and Architectures

Agent models provide the basis for defining an agent architecture. There exist different models (see section 2.4) among which the BDI model is popular for intelligent agents. It has its basis in philosophy and offers a logical theory which defines the mental attitudes of Belief, Desire, and Intention using a modal logic (Winikkoff *et al.,* 2001). The central concepts in BDI model are:

- Beliefs: Information about the environment.
- Desires: Objectives to be accomplished.
- Intentions: The currently chosen course of action.
- Plans: Means of achieving certain future world states.

1.3.1 Agent Architectures

A shift from theory to practice leads to a shift from model to architecture that is the construction of computer systems that satisfy the properties specified by an agent model. Kaelbling defines an agent archetecture as: "Specific collection of software (or hardware) modules, typically designated by boxes with arrows indicating the data and the control flow among the modules. A more abstract view of an architecture is a general methodology for designing particular modular decompositions for particular tasks" (Kaelbling, 1991).

Agent architectures can be categorized by the way they solve problems. The classic way to build agents is to build a knowledge base and then enable the agent to reason about the belief contained therein. This paradigm is known as symbolic AI (artificial intelligence). Agents built on this model of the world are known as *Deliberative Agents* or sometimes called symbolic reasoning agents. A deliberative agent architecture contains an explicitly represented symbolic model of the world (i.e. the knowledge base). Decisions in the agent are based on pattern matching and symbolic manipulation in the knowledge base. There are two types of problems with this type of approach (Jennings, 1995; Graham, 2001).

- How to translate the real world into an accurate, adequate symbolic description in a period of time rapid enough for that description to be useful.
- How to represent information about complex real world entities and processes and how to get agents to reason about that representation in time to be useful.

These problems have led researchers to develop what are now known as Reactive Architectures. In reactive architectures there is no symbolic reasoning. This idea is that intelligent behavior can be generated without explicit symbolic representation. Without the symbolic representation there is no need for the complex symbolic reasoning. In a reactive agent, intelligence is an emergent property of complex systems (Rodney, 1986). Intelligence emerges as a result of an agent's interaction with its environment. Real intelligence is situated in the real world and not in a disembodied representation as in theorem provers and expert systems (Rosenschein and Zlotkin, 1994). Many researchers have suggested that neither a completely deliberative nor completely reactive approach is suitable for building agents. They have argued the case for *Hybrid architectures.* A Hybrid architecture contains the components that are reactive and also components that are deliberative (Jennings and Wooldridge, 1995).

1.4 Single Agent Limitations

A single agent is an entity that performs a wide range of user-delegated information finding tasks with general knowledge. We believe that such centralized approaches have several limitations. A single general agent would need an enormous amount of knowledge to be able to deal effectively with user information requests that cover a variety of tasks. In addition, a centralized system constitutes a processing bottleneck and a single point of failure. Furthermore, unless the agent has beyond the state of the art learning capabilities, it would need considerable reprogramming to deal with the appearance of new agents and information sources in the environment. Finally, because of the complexity of the information finding and filtering task, and the large amount of information, the required processing would overwhelm a single agent (Sycara and Zeng, 1996).

Most researchers in AI to date have dealt with developing theories, techniques, and systems to study and understand the behavior and reasoning properties of a single cognitive entity. AI has matured, and it endeavors to attack more complex, realistic, and large-scale problems. Such problems are beyond the capabilities of an individual agent. The capacity of an intelligent agent is limited by its knowledge, its computing resources, and its perspective. This bounded rationality is one of the underlying reasons for creating problem-solving organizations. The most powerful tools for handling complexity are modularity and abstraction (Sycara, 1998).

1.5 Multi-agent System Characteristics Over Single Agent

Nikos defines the fundamental aspects that characterize multi-agent systems (MAS) and distinguishes it from a single-agent system. These fundamental aspects are given below (Nikos, 2003):

- **Agent design:** It is often the case that different agents which comprise MAS (multi-agent system) are heterogeneous and that they have different capabilities and behaviors that would lead to design differences in hardware or software. Agent heterogeneity can affect all functional aspects of an agent from perception to decision making, while in single-agent systems the issue is simply nonexistent.
- **Environment:** Agents have to deal with environments that can be either static (time invariant) or dynamic. Most existing AI techniques for single agents have been developed for static environments. In MAS, the mere presence of multiple agents makes the environment appear dynamic from the point of view of each agent.
- **Perception:** The collective information that reaches the sensors of the agents in MAS is typically distributed, i.e. the agents may observe data that differ spatially (appear at different locations), temporally (arrive at different times), or even semantically (require different interpretations).
- **Control:** Contrary to single-agent systems, the control in MAS is typically distributed (decentralized). This means that there is no central process that collects information from each agent and then decides what action each agent should take.
- **Knowledge:** In single-agent systems we typically assume that the agent knows its own actions but not necessarily how the world is affected by its actions. In MAS, the levels of knowledge of each agent about the current world state can differ substantially.
- **Communication:** Interaction is often associated with some form of communication. Typically we view communication in MAS as a two-way process, where all agents can potentially be senders and receivers of messages. Communication can be used in several cases, for instance, for coordination among cooperative agents or for negotiation among self-interested agents (competitive agents).

1.6 Multi-Agent Systems Motivation

In this section we will develop our understanding about the motivations of MAS by going through different perspective.

1.6.1 General Perspective

The motivation for the increasing interest in MAS research from the general perspective may include the following abilities (Chavez *et al*., 1997; Yanai, 1999):

- To solve problems that are too large for a centralized single agent to do due to resource limitations or the sheer risk of having one centralized system
- To allow for the interconnection and interoperation of multiple existing systems, e.g. expert system, decision support systems, etc.
- To provide solution to inherently distributed problems.
- To provide solutions which draw from distributed information sources.
- To provide solutions where the expertise is distributed e.g. in healthcare.
- To enhance speed (if communication is kept minimal), reliability (capability to recover from the failure of individual components, with graceful degradation in performance), extensibility (capability to alter the number of processors applied to a problem), and the ability to tolerate uncertain data and knowledge.
- To offer conceptual clarity and simplicity of design.

1.6.2 Information Gathering Perspective

Motivations for multi-agent systems from the information gathering perspective could be analyzed by looking at the qualities of agents that can benefit the tasks related to information gathering (Carrick and Qiang, 1999). Some of them are given below:

- Delegation: An agent can receive a query in which the data sources necessary to answer the query are not specified.
- Transparency: The user should not need to specify how the data is distributed among the information sources.
- Data-directed execution: Agents can notify users of updates or revised patterns in information sources.
- Communication: Agents can communicate with other agents to obtain information from resources to which they do not have access.
- Reasoning: An agent can optimize a given query to minimize communication, time and expense.
- Planning: An agent can formulate a plan to access appropriate resources to answer a query.

1.6.3 Information Processing and Problem Solving Prospective

In this section, we describe the multi-agent system from the intelligent information retrieval and problem solving perspective. Characteristics of this perspective include the following (Sycara *et al*., 1996):

- Distributed information sources: Information sources available on-line are inherently distributed. Furthermore, these sources typically are of different modalities. Therefore it is natural to adopt a multi-agent architecture consisting of many software agents specialized in different heterogeneous information sources.
- Sharability: Typically, user applications need to access several services or resources in an asynchronous manner in support of a variety of tasks. It would be wasteful to replicate agent information gathering or problem solving capabilities for each user and each application. It is desirable that the architecture support sharability of agent capabilities and retrieved information.

- Complexity hiding: Often information retrieval in support of a task involves quite complex coordination of many different agents. To avoid overloading the user with a confusing array of different agents and agent interfaces, it is necessary to develop an architecture that hides the underlying distributed information gathering and problem solving complexity from the user.
- Modularity and Reusability: One of the basic ideas behind the multi-agent-based approach is that software agents will be kept simple for ease of maintenance, initialization and customization. Another fact of reusability is that pre-existing information services, whose implementation, query language and communication channels are beyond the control of user applications, could be easily incorporated in problem-solving.
- Flexibility: Software agents can interact in new configurations “on-demand", depending on the information requirements of a particular decision making task.
- Robustness: When information and control is distributed, the system is able to degrade gracefully even when some of the agents are out of service temporarily. This feature of the system has significant practical implications because of the dynamic and unstable nature of on-line information services.
- Quality of Information: The existence of (usually partial) overlapping of available information items from multiple information sources offers the opportunity to ensure (and probably enhance) the correctness of data through cross-validation. Software agents providing the same piece of information can interact and negotiate to find the most accurate data.
- Legacy Data: Many information sources exist prior to the emergence of Internet-based agent technology. New functionalities and access methods are necessary for them to become full-fledged members of the new information era. Directly updating these systems, however, is a nontrivial task. A preferable way of updating is to construct agent wrappers around existing systems. This agent wrapper approach offers much flexibility and extensibility. Practically speaking, it is also easier to implement since the internal data structure and updating mechanism of the legacy information systems do not need to be modified.

1.7 Multi-Agent System Overview

Traditionally, research into systems composed of multiple agents was carried out under the banner of *Distributed Artificial Intelligence* (DAI), and has historically been divided into two main camps (Shoham, 1997): *Distributed Problem Solving* (DPS) and *Multi-Agent Systems* (MAS). More recently, the term “multi-agent systems” has come to have a more general meaning, and is now used to refer to all types of systems composed of multiple (semi-)autonomous components. Distributed problem solving (DPS) considers how a particular problem can be solved by a number of modules (nodes), which cooperate in dividing and sharing knowledge about the problem and its evolving solutions. In a pure DPS system, all interaction strategies are incorporated as an integral part of the system. In contrast, research in MAS is concerned with the behavior of a collection of possibly pre-existing autonomous agents aiming at solving a given problem.

A MAS can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver (Georgeff and Rao, 1995). These problem solvers—agents—are autonomous and may be heterogeneous in nature. When a group of agents in a multi-agent system share a common long-term goal, they can be said to form a team. Team members (or teammates) coordinate their behaviors by adopting compatible cognitive processes and by directly affecting each others' perceptory inputs including via communicative actions. As such, characteristics of MAS are:

- Each agent has incomplete information, or capabilities for solving the problem, thus each agent has a limited viewpoint.
- There is no global system control.
- Data is decentralized.
- Computation is asynchronous.

In order to obtain coherent system behavior, individual agents in a multi-agent system should not only be able to share knowledge about problems and solutions, but should also reason about the processes of coordination among other agents. Without coordination all the benefits of decentralized problem-solving can quickly vanish. As in a multi-agent system, there is no possibility of global control, globally consistent knowledge, global success criterion, or even a global representation of the system, and therefore the task of coordination can be quite difficult. The coordinating task raises issues in communication, interaction, cooperation, and coherence.

1.7.1 Coordination

A distinguishing feature of multi-agent system is the fact that the decision making of the agent can be distributed. This means there is no central controlling agent that decides what each agent must do at each step, but each agent is to a certain extent responsible for its own decisions. The main advantages of such a decentralized approach over a centralized one is efficency due to asynchronous computation and robustness in the sense that the functionality of the whole system does not rely on a single agent. In order for an agent to be able to carry out thier actions in a distributed fashion, appropriate additional coordination mechanisms must be additionally developed. Note that coordination can be regarded as the process by which the individual decision of the agents result in good joint decisons for the group. A typical situation where coordination is needed is among cooperative agents that form a team, and through this team, they make plans and pursue common goals.

1.7.2 Communication

Multi-agent interaction is often associated with some kind of communication. Agents' communication involves several levels of abstraction. On the lower, “network” level, it must be made sure that the messages that are sent among the agents arrive safely and on time at their destination. On an intermediate, ‘language’ level, there must be a basic set of language primitives and a standardized format for exchanging these primitives, so that agents that speak the same language can easily understand each other. On a higher, 'application' level, there must be effective use of communication for solving standard multi-agent problems, like coordination or negotiation By nature, multi-agent systems are generally distributed, making interaction more difficult. Interaction in a multi-agent system involves two key components. First is the language or interaction protocol, and second is the use of performatives.

In a conversation-centric system, the actions of an agent are driven by the communications it has with other agents in the system. When dealing with distributed agents, it is important to develop a common language that any agent can understand. The use of a common language ensures that any new agent can receive a message, and based on the language protocol in use, extract the information it requires. This is accomplished through a common message format. A common format allows agents to interface with other agents regardless of the agents' internal structure. Once a common communication protocol has been established, performatives must be developed that can give receiving agents direction or direct agent actions towards a system goal. As such, performatives are the speech-act components of the language (Bratman, 1992). Performatives are specific to each system and are dependent on the functions and goals of that system.

1.8 Challenging Issues in Multi-agent Systems

The transition from single-agent systems to MAS offers many potential advantages, but they also face many difficult challenges. We present problems inherent in the design and implementation of MAS (Sycara, 1998; Sandholm and Lesser, 1995; Nicholas *et al*., 1998):

- How to formulate, describe, decompose, and allocate problems and synthesize results among a group of intelligent agents?
- How to enable agents to communicate and interact? What communication languages and protocols to use? What and when to communicate?

- How to ensure that agents act coherently in making decisions or taking action, accommodating the non-local effects of local decisions and avoiding harmful interactions?
- How to enable individual agents to represent and reason about the actions, plans, and knowledge of other agents in order to coordinate with them? How to reason about the state of their coordinated process (e.g. initiation and completion)?
- How to recognize and reconcile disparate viewpoints and conflicting intentions among a collection of agents trying to coordinate their actions?
- How to effectively balance local computation and communication? More generally, how to manage allocation of limited resources?
- How to avoid or mitigate harmful overall system behavior, such as chaotic or oscillatory behavior?
- How to engineer and constrain practical MAS systems? How to design technology platforms and development methodologies for MAS?
- How to handle the distributed perceptual information? How to enable agents to maintain consistent shared models of the world?
- How to implement decentralized control and build efficient coordination mechanisms among agents?
- How to enable agents to form organizational structures like teams or coalitions? How to assign roles to agents?
- How to define a systematic methodology enabling designers to clearly specify and structure their applications as multi-agent systems?

1.9 Agent Systems and Applications

The intelligent agent concept is a new paradigm for software application development and agent-based computing has been hailed as 'the next significant breakthrough in software development'

(Sargeant, 1992). According to Jennings and Wooldridge (1995), there are several dimensions along which agent applications could be classified, like type of the agent, technology used to implement the agent or by the application domain itself. Following (Jennings and Wooldridge, 1995) we briefly define applications by domain type.

1.9.1 Industrial Applications

Agent technology was first used in the industrial domian as early as 1987. Today agents are being applied in a wide range of industrial applications.

Process control: Process control is a natural application for intelligent agents and multi-agent systems,the best known of these is ARCHON (Jennings and Wooldridge, 1995).

Manufacturing: YAMS (Yet Another Manufacturing System) was described by Parunak (1987), which applies the Contract Net Protocol (Smith,1980) to manufacturing control

Air Traffic Control: A sophisticated agent-based air traffic control system was developed by Kinny *et al.* (1996) that is known as OASIS. In this system, agents are used to represent both aircraft and the various air-traffic control systems in operation

1.9.2 Commercial Applications

Agent technology is also being used in commercial applications, particulary in information managment like information filtering and information gathering.

Information Management: There are many projects which have been developed under the title of agent-based infomation management, e.g. Maxims, Newt, etc. Maes (1994) developed an electronic mail filtering agent called Maxims that learns to prioritize, delete, forward, sort, and archive email messages on behalf of the user. *Newt* was also developed by (Maes, 1994) that is an example of an agent-based internet news filtering.

1.9.3 Medical Applications

Agent technology has also been actively deployed in new applications in the healthcare industry. Early applications are in the areas of patient monitioring. Following this for this thesis, we will focus on the healthcare domain and we will review the issues, problems, and motivations on how agent technology can be useful for healthcare applications.

"Healthcare at all levels - local, regional, national and international - is a vast open environment characterized by shared and distributed decision making and management of care, requiring the communication of complex and diverse forms of information between a variety of clinical and other settings, as well as the coordination between groups of healthcare professionals with very different skills and roles" (John and Moreno, 2003). The main focus of healthcare software systems is to operate effectively in this environment, in order to meet the information needs of patients and health care providers. Practitioners in healthcare environments require just in time and error free information, such that recommendations or decisions offered by the software systems are secured and trustworthy (Fox *et al.,* 2000). There is a growing interest in the application of agent-based techniques to problems in the healthcare domain (Weiss, 1999; Wooldridge, 2000). We argue that multi-agent systems are indeed an interesting tool capable of solving those problems, since the usual properties of intelligent agents match quite precisely with the needs of the healthcare domain (basically with the requirement of having autonomous intelligent proactive collaborative entities in a distributed environment) (John and Moreno, 2003).

1.9.3.1 Characteristics of Healthcare Domain Problems

Here, we describe briefly some important characteristics of problems in the healthcare domain.

- It is very usual that the knowledge required to solve a healthcare problem is spatially distributed in different locations. For instance, the problem of patient scheduling requires the scheduling of the different tasks to be performed on a hospitalized patient (Decker *et al.,* 1998; Kumar *et al.,* 1998; Marinagi *et al.,* 2000).

- Healthcare problems are quite complex, and finding standard software engineering solutions for them is not straightforward. For instance, coordinating the process of organ transplant in a country is not an easy task (Aldea *et al.,* 2001).
- In the last few years there has been a shift in healthcare practice towards health care promotion, shared patient-provider decision-making and managed care, creating an increased demand for information and online services (Silverman *et al,.* 1998; Shankararaman *et al.,* 2000). To meet these needs the software systems must be proactive in anticipating the information and knowledge needs of users and deliver it in a timely manner, support synchronous and asynchronous communication, and facilitate collaborative decision making between the various individuals involved in the process of managing and delivering healthcare services.
- There is a great amount of medical knowledge available on the Internet. It is necessary to provide ways of accessing the most relevant information as easily, flexibly and timely as possible. This access to healthcare information is necessary both for healthcare practitioners and for patients. In both cases it is important to receive appropriate information from useful and reliable sources in a proactive way, without having to devote time and effort to look for, analyze, evaluate or filter it.

1.9.3.2 Appropriateness of Agent Technology to Health Care Problems

Following our discussion on the characteristics of the healthcare domain, we will now describe how agent technology can be adequately applied to the healthcare domain. The components of a multi-agent system may be running on different machines, located at many different places. Each of the agents may keep part of the knowledge required to solve the problem, therefore, multi-agent systems offer a natural way of attacking inherently distributed problems. One of the main properties of an intelligent agent is sociability. In that way they can engage in complex dialogues in which they can negotiate, coordinate their actions and collaborate in the solution of a problem (e.g. different units of a hospital may collaborate in the process of patient scheduling (Decker *et al.,* 1998; Kumar *et al.,* 1998; Marinagi *et al.,* 2000). When a problem is too complex to be solved in a single system, it is usual to decompose it into sub-problems (which will probably not be totally independent of each other). Multi-agent systems can handle the complexity of solutions through decomposition, modeling and organizing the interrelationships between components. Agents can also be used to provide information to doctors and patients by retrieving information from different sources, analyzing the obtained data, selecting the information in which a user is especially interested in, filtering redundant or irrelevant information, and presenting it to the user with an interface adapted to the user's preferences. The proactive nature of agents allows them to perform tasks that may be beneficial for the user, even if he/she has not explicitly demanded those tasks to be executed. Their inherent autonomous nature also allows each agent to make its own decisions, based on its internal state and the information that it receives from the environment (e.g. each unit of the hospital may keep its private data, or each hospital may use a different policy to rank the patients that are waiting for an organ transplant).

1.10 Research Focus

1.10.1 Problem Statement

The area of multi-agent systems is very vast and multi-dimensional. In this chapter, we have developed our understanding with regards to the motivations which underpin the development of multi-agent systems, its challenges, problems and other related issues. Literature on intelligent agents and multi-agent has revealed that many researchers have tried to address the problems related with this technology but being a complex and comparatively new paradigm of software-problem solving, there are many issues which impede the design and implementation of MAS (see section 1.8). Therefore, the focus of this research is to come up with the state-of-the-art strategies and techniques for designing and implementing cooperative multi-agent system applications and to develop a multi-agent based system for *Intelligent Healthcare Knowledge Assistance and Capitalization ((IHA), Note: for simplicity we call it IHA i.e. intelligent healthcare assistant).* Given this focus, we have identified key problems that we aim to address in this thesis.

Ineffective individual single agent model: The individual intelligent agent model provides the basis for any agent architecture. The most popular model for individual agent specification is BDI. However, the concepts we have found useful for agent development within BDI systems do not necessarily match the concepts proposed in theory. Neither are they exactly the concepts which have been implemented in systems such as JACK (Winikoff *et al*., 2001). Sycara (1998) describes that agent having BDI model specification are regarded as Deliberative whereas we argue that to mimic human behavior, there is a need for a model which provides the basis for hybrid behavior in an autonomous individual agent architecture.

Inadequate individual agent architecture: Sophisticated individual agent reasoning can increase MAS coherence and define a problem at the individual agent level to enable individual agents to represent and reason about the action, plans, and knowledge of other agents in order to coordinate with them (Sycara, 1998; Nicholas *et al.,* 1998). However, we argue that present Deliberative and Reactive architectures do not suit to mimic human-like behavior. Both capabilities are essential but an additional attribute called motivation is perhaps also necessary. Beaudoin (1994) has identified the need for a rich representation of motivators when dealing with complex human-like environments.

Insufficient cooperative multi-agent system model: The benefit of decentralized multi-agent problem-solving can not be capitalized upon if multi-agent systems do not enforce proper coordination mechanisms. In a multi-agent system, there is no possibility of global control, globally consistent knowledge, global success criterion, or even a global representation of the system. Therefore, the task of coordination can be quite difficult. The coordinating task raises issues in communication, interaction, cooperation, and coherence (Chauhan, 1997). Presently, these issues are not adequately dealt with in many existing multi-agent cooperative systems which lead to increased communication and resource usage overheads.

Insufficient multi-agent communication language: In MAS a communication language is necessary to carry out different functionalities and mechanism like coordination, cooperation, knowledge sharing, and capability description. Sycara, Jennings and Wooldridge (1995) define the multi-agent communication problem as an impedance for the effective design and implementation of MAS main problems concerning multi-agent communication include “How to enable agents to communicate and interact? What communication languages and protocols to use? What and when to communicate?” (Sycara, 1998; Nicholas *et al.,* 1998).

Unsuccessful explication or formalism of agent methodologies: One of the major technical impediments to the widespread adoption of agent technology is the lack of a systematic methodology enabling designers to clearly specify and structure their applications as multi-agent systems (Sycara, 1998). This means that most existing applications have been designed in a fairly ad hoc manner – either by borrowing a methodology (typically an object-oriented one) and trying to adapt it to the multi-agent context or by working without a methodology and designing the system based on intuition and past experience that is unsatisfactory. So, what is required is a systematic means of analyzing the problem, of working out how it can be best structured as a multi-agent system, and then determining how the individual agent can be structured.

Overwhelming volume of healthcare multi-modal knowledge, and inadequate knowledge usage and capitalization: In the healthcare industry, multi-modal knowledge represents the different forms of knowledge. These can be categorized as:

- cases- storing experiential knowledge,
- scenarios-storing tacit knowledge, and
- documents-storing explicit knowledge.

In the current situation where the volume of healthcare information and knowledge that is being transacted daily is simply overwhelming, healthcare enterprises are constantly under pressure to ensure that these information and knowledge are effectively and efficiently utilized and managed. These lead to the following problems:

1. The lack of just-in-time knowledge assistance for critical and non-critical healthcare situations from multi-modal (cases, scenarios and documents) and distributed knowledge sources in a robust and autonomous way. Even most medical knowledge-based systems, although built with the aim of delivering accurate and timely results, are often suitable only to address non-critical healthcare situations from a single modal knowledge source. These do not command the trust to handle critical healthcare problems in providing just in time knowledge assistance in autonomous and robust fashion;
2. The insufficient capitalization of multi-modal knowledge as a structured organized and distributed healthcare enterprise memory (HEM). This is because the knowledge within a local HEM may not be sufficient to address the required needs and requires state of the art strategies and techniques to efficiently organize and structure multi-modal knowledge.

1.10.2 Problem Solution

Given the above problem definitions, those can be addressed by the following problem solution:

- By developing an agent model that better reflects human behavior, so that agents could be the right entity to work on behalf of humans in carrying out specific tasks to alleviate the cognitive load. In our view, it is not fully possible for an agent to mimic human behavior if it just conforms to deliberative or reactive behavior. That is why we have taken the hybrid approach. The agent model proposed in our work is based on the agent definition (see section 4.2) which is a composition of different properties and attributes defined at different layers and includes other behavioral properties that would add value to mimic human behavior.

- The design of hybrid sophisticated autonomous individual agent generic architecture. We, take a holistic approach and this involves the following steps: (i) Defining agent with behavioral attributes/properties, (ii) Classification of the properties into layers according to their abstract level functionalities, (iii) Designing an agent model based on the layered classification in order to have high-level view of the agent main modules, and (iv) Decomposition and interaction of the agent high level main modules into agent functional work flow.

- By developing of a DBC (Demand Based Cooperation) model for multi-agent systems. The basis for the DBC model emerges from the cooperative nature of problem solving. The DBC model addresses issues related to cooperation, coordination and communication of multi-agent system to improve the overall performance of the system. Every cooperative multi-agent system must follow the protocols related to cooperation, coordination and communication which are described in the model on which the multi-agent system is built. We argue that to achieve certain goals of the system in multi-agent system, it is not necessary for all agents to be active and have cooperation, coordination and communication and if all agents become active they would cause penalty in terms of system performance, communication overhead and coordination management. Our approach for cooperation in multi-agent systems is based on the *"on demand cooperation"* concept, which is defined as *real time agents' selection for a team that suits best to the demand for achieving certain goal"*. In this manner, those agents who have been selected to achieve a goal would exhibit cooperation, coordination and communication.

- By defining a framework of a KDL (Knowledge Description Language) communication language that adheres to the protocols and policies defined for communication in the DBC model for multi-agent systems. KDL is structured in XML frame that provides speech act performatives defined in KQML. These frames encapsulate semantically rich data for knowledge exchange and sharing, support agent capability description representation and provide a easy and flexible way to parse, interpret and process communication messages among agents.

- By designing of an agent-oriented methodology for the analysis and design of multi-agent systems which is completely based on the agent–oriented concept that incorporates organizational structure, organizational model, multi-agent system architecture (design), services, goals, responsibilities, plans/actions and interaction protocols.

- By developing an Intelligent Healthcare Assistant (IHA) framework consisting of: (a) a HEM (healthcare enterprise memory)—a healthcare knowledge web—which provides access paths to diverse knowledge sources; and (b) agent-mediated intelligent access to, and procurement of, heterogeneous knowledge by approximate matching of resources, content navigation, and content correlation. IHA's focused knowledge search and navigation is grounded in five fundamental principles: (i) it employs specific functionally-autonomous knowledge retrieval and procurement agents for each constituent repository; (ii) it employs a common ontology modeling the knowledge objects; (iii) it collects knowledge by leveraging a medical ontology that assists knowledge matching; (iv) it populates the HEM from only those sources that need to be accessed for relevant content; and (v) it ensures inter-agent communication for agent collaboration to traverse the HEM for 'holistic' knowledge retrieval. Interaction with a HEM is facilitated by the IHA, whereby the user's knowledge needs are specified as a Knowledge Specification (KS)—akin to a query. The inherent medical ontology allows for the expansion of the KS.

1.10.3 Contribution of this Thesis

The contribution of this thesis can be visualized in five layers (see figure 1.1). We will define each layer as follows:

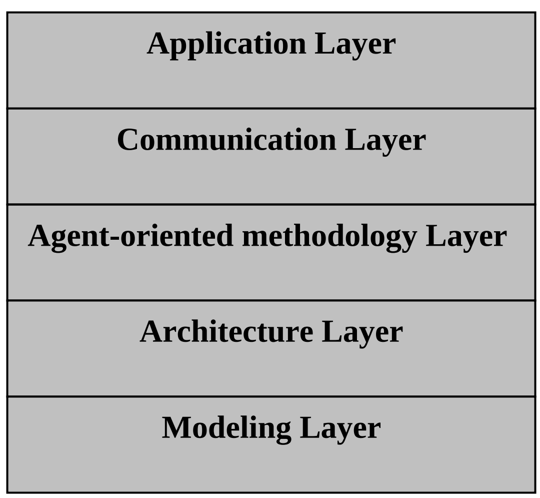

Figure 1.1 Contribution of the thesis

Modeling Layer: Our contribution at this layer is related to the modeling of intelligent agents and multi-agent systems. These models help in designing architectures for both single agents and multi-agent:

- First model: *Hybrid Intelligent Autonomous Agent (IAA) model.* This model provides the basis for the generic intelligent agent architecture which aims to mimic human behavior.
- Second model: *DBC cooperative multi-agent model.* This second model provides the basis for multi-agent system workflow, interaction and communication.

Architecture Layer: The contribution at this level is centered around the designing for single agent and multi-agent systems architectures. The contributions at this layer are:

- Hybrid Intelligent Autonomous Agent (IAA) Generic Architecture which is based on the Hybrid IAA model. This architecture provides a platform to build intelligent agents that are different in nature (in terms of their job functions).
- IHA multi-agent system architecture that is based on *DBC cooperative multi-agent model* and deploys agents based on the IAA generic architecture. Note that in IHA multi-agent system architecture, it would comprise of a number of single agents.

Agent-oriented methodology Layer: Our contribution at this layer is an Intelligent Autonomous Agent(IAA)-oriented methodology for the analysis and design of agents involved in a multi-agent system.

Communication Layer: The contribution at this layer involves the communication language to be used in a multi-agent system. The outcome of this layer is a KDL language which is complete with speech acts, knowledge description and capability description of agents. KDL is a content level language defined with KQML for DBC cooperative multi-agent communication protocols and IHA knowledge description.