IHK: Intelligent Autonomous Agent Model and Architecture towards Multi-agent Healthcare Knowledge Infostructure

Zafar Hashmi[1], Somaya Maged Adwan[2]

[1]*Metavonix IT Solutions*
Smart Healthcare Lab,
Washington DC, USA
Zafar.hashmi@metavonix.com

[2]*Department of Electrical Engineering,*
University Malaya, Malaysia (UM),
Kuala Lumpur, MALAYSIA
somi@siswa.um.edu.my

ABSTRACT

Intelligent autonomous agents is an emerging technology that excels the applications, having complex distributed nature and requiring communication of complex and diverse forms of information, when deployed in multi-agent system setting. As healthcare is a complex domain and distributed in nature, scientist have started to deploy intelligent agents and multi-agent technologies indifferent healthcare information and decision support systems. We argue that efficacy of multi-agent system and performance of multi-agent system architecture in any domain, in particular healthcare domain heavily depends on a single agent architecture and model that mimic human behavior. In this paper we present, Hybrid Intelligent Autonomous Agent (IAA) Model, Hybrid Intelligent Autonomous Agent (IAA) Architecture, and based on these two fundamental building blocks, designing and development of an Intelligent Multi-agent Healthcare Knowledge Infostructure (IHK)

Keywords
Intelligent agents, architecture, multi-agent system, Healthcare Knowledge Management

1.0 INTRODUCTION

As a matter of fact, nature of healthcare as a whole is a complex wide open environment leading to shared and distributed decision making, communication of different types of information between clinical and other healthcare settings, and coordination among different healthcare practitioners (John, Moreno, 2003). Such complex domain (healthcare) has attracted intelligent agents and multi-agents researcher to solve healthcare problem using new state of the art technology that itself (in nature) has a great match with healthcare complex nature. Intelligent agents in multi-agent setting are being deployed in healthcare systems vis-à-vis healthcare information systems, clinical decision support systems etc. 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, Moreno, 2003). The functioning and performance of multi-agent system and its architecture heavily depend on the single agent architecture and its model that mimic human behavior. To enhance the efficacy of healthcare multi-agent systems, we present in this paper (i) Hybrid Intelligent Autonomous Agent (IAA) Model, (ii) Hybrid Intelligent Autonomous Agent (IAA) Architecture, and to see how this single agent model and architecture could best suit multi-agent system, an Intelligent Healthcare knowledge Assistance and Capitalization (IHA) Multi-agent system based and developed on these building blocks will be put for-ward towards the end.

2.0 HYBRID INTELLIGENT AUTONOMOUS AGENT (IAA) ATTRIBUTES

It has been found in literature (Bradshaw, 1997) that there is no standard definition of an intelligent agent. Therefore, it is useful to define our understanding of intelligent agents before applying them into practice. In our view, an intelligent agent is a software entity which mimics human behavior which may consist of five layers whereas each layer has its specific attributes to make an agent intelligent and powerful. These layers are as follows: Environmental Interaction, Motivational Control State, Mental State, Cognitive Functions, and Flexible Control.

1) Environmental Interaction: This layer defines the attributes of the agent which are used to interact with the environment. These attributes are: *Perception:* This refers to the acquisition and interpretation of data about environmental states and events; Action: This is the agent output including actions that changes the state of the environment; Communication: This refers to the interaction with other agents in the environment through conversation; Event: An event is a significant occurrence. Events are often extracted from percepts, although they may be generated internally by the agent. An event can trigger new goals, cause changes in information about the environment, and/or cause action to be performed immediately. **2) Motivational Control State:** The motivational control state refers to motivations that move an agent towards a de-sired physical/mental state in light of agent beliefs and concerns, i.e. a subclass of information structures with dispositional powers to determine action (both internal and external), and which subsumes desires, goal , intentions and wishes.

3) Agent Mental State: The agent maintains a representation of the state of its environment and its situation in the environment that may include: *Beliefs:* These are what the agent holds to be true about its environment and itself; Desires: These are the agent's current goals for itself and the environment; Intentions: These are actions plans or other tasks to which the agent is committed; Knowledge: This is a special class of beliefs, refereeing to general laws, ontologies, rules, and etc. rather than specific situations.

4) Cognitive Functions: Agent can implement cognitive functions over their mental states such as: *Reasoning:* Making inferences on the basis of its beliefs, knowledge etc; Decision-making: making rational choices, often involving uncertainty; Planning: Constructing collection or sequences of actions to achieve its goals; Scheduling: Adap-tively controlling its plans and actions as the environment changes; Learning: Storing and recalling past situations and solutions to problem goals.

5) Flexible Control: An agent also needs to maintain flexible control of its behavior in response to circumstances, achieving a balance of: *Deliberative control:* This is the execution of goal-directed, coordinated actions carried out over time; Reactive control: Responding to situations and events even if they are unexpected; Autonomy: This is the capacity to independently plan, make decisions and act based on its goals and beliefs; Adaptivity: This is to change its behavior according to the constraints for the given task requirement.

3.0 HYBRID INTELLIGENT AUTONOMOUS AGENT (IAA) MODEL

The community of agent researchers believes that agents perform tasks on behalf of humans. In line with this belief, we develop an agent model that reflects human behavior, so that agents could effectively assist in alleviating the cognitive load of humans. In our view, it is not possible for an agent to mimic human behavior if it is just deliberative or reactive and does not include motivational control (through senses). That is why we have taken the hybrid approach which also takes into ac-count the motivations of the agent through sensors. This model consists of 5 levels which are wrapped with sensors. Each level has its own set of attributes which are based on the IAA definition. ***Brain:*** Here, the brain is an analogy of the human brain. Its attributes corresponds to the three layers in the IAA definition (i) Cognitive Functions (ii) Flexible Control (iii) Motivational Control state. It can perform reasoning, decision making, planning and scheduling and can have motivations to achieve/not achieve its goals. It is responsible to behave deliberatively or reactively based on the perceptions. Autonomy and adaptiveness are also maintained at this level. To perform all these functions, the brain must be equipped with environ-mental and common sense knowledge that is defined in IAA 'agent mental state' layer.

Knowledge Base: The knowledge base in our model is responsible to maintain the attributes defined in the Agent Mental State layer that is: beliefs, desires, intentions and knowledge (environmental, common sense, etc). ***Actions:*** The actions level is responsible for all kinds of actions both reactive and deliberative. An action is something an agent does, that affects its environment (Maes ,1994). In their simplest form, actions are atomic and instantaneous, can either fail or succeed, can be durational (encompassing behavior over time) and can produce partial effects. Actions are also part of an agent's interface to the environment. This level (actions and events) implements partially the first layer of agent definition that is 'environmental interaction' layer. ***Communication:*** This is the ability of an agent to exchange its views, information and commands in the environment with other agents irrespective whether it is a part of multi-agent system or not. In our view, even if there is one only agent deployed in the system, it must still have this ability so that it can readily communicate with other agents in future.

The communications attribute in the 'environmental interaction' layer of our IAA definition is part of this level where communication protocols and communication ontology mechanisms are controlled. ***Sensors:*** In our model sensors are wrapped over all levels, i.e. affecting all the attributes of each level. Sensors implement the perceptions of an agent where perception refers to the ability to acquire and interpretation of data about environmental states and events. Motivations are also deployed at this level as a function of senses. ***Neighborhood:*** In our view, an agent must be able to recognize its neighborhood if it becomes a part of the system where more than one agents are deployed. This level reduces the load of the Brain level. In this level an agent keeps enough knowledge to recognize other agents when it is deployed in a multi-agent system.

3.1 Agent Execution Cycle

We would now define hybrid IAA model agent execution cycle. Execution cycle is defined (see Figure 1) in the following steps. This execution cycle is to give an understanding that how our hybrid IAA model improves over other related agent models (Winikoff , Padgham, and Harland, 2001). (1) Sensors are activated to give perceptions that produce events or trigger communication (that in turn produce some events for new information). (2) Events are scheduled according to the priority. (3) Events and communication yield reflexive actions or deliberative actions. (4) Beliefs are updated with new information. (5) Goals are updated, including current, new and completed ones based on reasoning.(6) Motivation is set to high to achieve new goal (it goes to low automatically based on situation). (7) Plans are chosen based on rational decision-making. (8) The chosen plan (intention) is expanded to yield actions. (9) If a plan fails the agent tries another plan. (10) If all plans in the plan library could not achieve the desired goal then re-planning is carried out.

4.0 HYBRID INTELLIGENT AUTONOMOUS AGENT (IAA) ARCHITECTURE

The novelty of our approach in designing the IAA architecture is that the design approach is top-down and consists of multiple levels where each levels' output is taken as the input for the next level IAA is a hybrid intelligent autonomous agent architecture (see Figure 2) that it is neither purely deliberative nor purely reactive. Its hybrid nature allows it to perform both in deliberative and reactive manner. Additionally, state of the art communication and coordination functionalities make it collaborative agent as well. In the following, we define each of the IAA's modules. ***Controller:*** The Controller module in IAA is a key module which performs most of the cognitive functions of the agent (reasoning, rational decision-making, etc.) and deals with the control flow of the agent functionalities. Agent initialization is a part of its responsibilities. Initialization in our architecture is a very simple process, i.e. when the agent is started (initialized), the controller checks the dispatcher for incoming or out-going messages (or opens ports to receive or send messages).

This simple initialization process allows the IAA agent to turn itself off (sleep) if no job is to be performed (after a certain time) and to turn on (wake) if new job is to be done. It is also responsible for the autonomy of the agent (i.e. by having its own controller and running under its own thread). Besides these, the controller deploys two sub-modules for making the agent reactive and deliberative in nature. The main reason for separating the reactive module from the deliberative module is to stay true to the modular philosophy behind the architecture. These two modules have quite separate roles: ***Deliberative Module:*** This module takes the traditional symbolic AI approach to rational agency that is to treat control states as semantically rich information structures to be manipulated according to Newell's principle of rationality (if an agent has knowledge that one of its actions will lead to one of its goals, then it will select that action) (Newell, 1982). This module performs all the logical reasoning and rational decision making involved in the architecture. It works with the knowledge base which contains (as one of its part) an explicitly represented symbolic model of the world (the environment it is designed for) that agent can use to formulate plans in order to achieve the goal states. It exhibits Event-and Goal-driven Behavior of the agent.

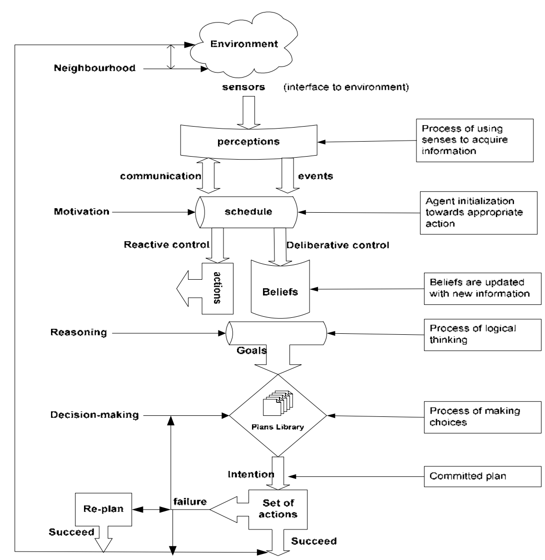

Figure 1 IAA Hybrid Model execution cycle

Reactive Module: For the reactive module, there is no central symbolic model of the world and no complex symbolic reasoning involved. Rather the agent would work in a hard wired stimulus manner. The idea behind the reactive module is that some intelligent behavior can be generated without the symbolic representation (certain sensor information always result in a specific action being taken). This module is responsible for the behavior-based nature of the agent architecture. ***Knowledge Base:*** The IAA knowledge base is like a knowledge warehouse that contains all kinds of mental state knowledge (belief, environmental knowledge, goals (desires), and intentions) in its sub-domains. It contains the knowledge needed for deliberative behavior in a representation required for it. Knowledge required for reactive behavior is also a part of this knowledge base. It also keeps the knowledge to work with the re-planner module (defined below) so that it could prepare plans for the goals (if all the defined plans failed for the specific goal(s)).

Goal Module: This module is very simple but very important for IAA. It contains all the goals along with their priority values. This priority value is used by the event scheduler to schedule the on going goals. The IAA architecture does not impose any specification for the priority function. It could be any value or a result from any function. This depends on the application or system that is using IAA architecture. ***Plan Library:*** In IAA, the plan library consists of (i) a set of plans for each goal along with (ii) its priority value which is used by priority index (PI) to arrange them in a sorted manner. Putting the plans in a sorted order in PI increases the response time of the architecture by reducing the reasoning and rational decision-making at the plans level while the agent is a part of a working system or application. The Plan library is dynamic in nature in a sense that it works also with the re-planner module to receive new plans for the on going goal and store them with PI. Its basic working principle is based on thee idea that the system needs to commit to the plans and sub-goals it adopts but must also be capable of reconsidering these at appropriate (crucial) moments.

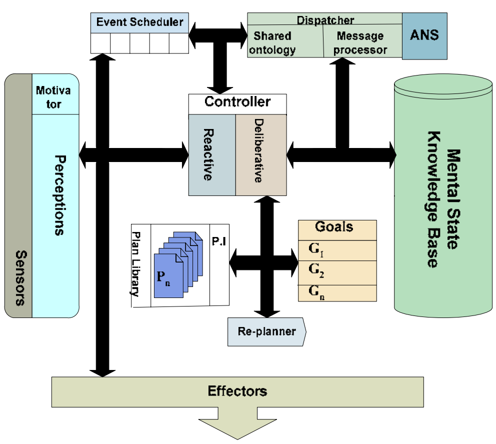

Figure 2 IAA Architecture

Re-planner: The re-planner is a planning module that is invoked when plans associated with the certain goal in the plan library have failed. The module works with four other modules i.e. knowledge base, deliberative controller, goal module, and plan library to find a new plan for the on-going goal. The technique of creating new plans is based on a searching mechanism that uses forward and backward chaining techniques to find the most appropriate set of actions for new plans in an attempt to shift the ongoing goal status from failure to success.

This technique has some resemblance to genetic algorithm where optimal solutions are searched from the search space. Every action must have some evaluation function so that it could be evaluated towards the success of the goal(s). For example let us assume that the knowledge base contains knowledge about different actions in branching representation where every node represents an action and it is connected with other node If action is evaluated to be successful then the next action is chosen to be a member of plan for ongoing goal. Let us say that a set of nodes {A,C,F,R,K} are evaluated as successful actions then they would be put in the plan in the same sequence and will then be executed through Effector module. If the goal is achieved after the execution of these set of actions, then they are included into a plan library as an accepted plan for that specific goal. The plan library stores, sorts and record it in PI. ***Event scheduler:*** An event is a significant occurrence which is often extracted from percepts and may be generated internally by the agent. In our work events can trigger reactive or deliberative behavior of the agent based on the nature of events. Event can be set to active state by percepts or by communication messages which inurn triggers the goals (deliberative) or actions (reactive) to be performed. Event scheduler schedules the event based on the priority. If a goal is selected to be achieved and during this process another event occurs that corresponds to a high priority value than the on-going goal, the on-going goal is transferred to a pending action queue (a part of Effector) and the new goal (deliberative) or action (reactive) of high priority is carried out. After executing of the higher priority event, the pending goal is completed. ***Effector:*** All the actions are carried out by the effector. The Effector is set into operation when there are any plans from plan library to be executed or any reflexive action to be performed. Each action from a specific plan is tagged with an id in the effector so that if this plan is to put into the pending action queue, a sub-module of the effector which keeps the pending goal along with their plan and actions can track it easily and put it back into execution mode. Effectors also contains response action queue that is used to keep the action that are waiting for response from the outside environment to achieve their goals. ***Dispatcher:*** The Dispatcher deals with network protocols and handles incoming and outgoing messages. It contains two message queues, one is for incoming messages and other is for outgoing messages. It waits for incoming messages and places them into their respective queues which in turn are passed to the message processor. It also takes messages which are to be sent to other agents by first, checking with the ANS (agent name server) to obtain port number/IP address of the agent to which the message is to be sent and then by establishing a connection to the target agent. Once the connection is made message is then sent.

Shared Ontology: From the artificial intelligence perspective ontology generally means the definition of a standard vocabulary and body of knowledge of some domain typically built using the defined standard vocabulary (Chandrasekaran, , Josephson, and Benjamins, 1999). Simply put, ontology is a specification of a conceptualization, aiming to de-scribe concepts and relationships between concepts. As exchanging messages among the agents is only part of the solution of communication problems, agents need to have shared ontology to exchange messages with less ambiguity (Aarsten, Brugali, and Vlad, 1996). In IAA shared ontology the generic structure (templates) maintains all communication messages and the concept definitions for each specific domain. These messages are built using agent communication languages such as KQML, etc. These generic messages structures from the shared ontology are then taken by a message processor to generate the appropriate communication message.

ANS (Agent Name Server): The purpose of the ANS is similar to most name servers such as DNS (Domain Name Server). The idea is that a new agent will register its existence with an ANS. A registration would consist of the new agent port number/IP address and a host name. Once the agent is registered, any other agents wishing to communicate will first contact the ANS to determine if the recipient is currently active. If the recipient is active the ANS will respond with the address and further communications will be carried on directly between the agents concerned. In other words the ANS helps agent to recognize their neighbors in a multi-agent system. ***Motivator:*** Work by Beaudoin (Beaudoin, 1994) has identified the need for a rich representation of motivators when dealing with complex human environments. The Motivator module is used to refer to motivational control states that move an agent towards a de-sired physical/mental state in light of agent beliefs and concern, i.e. subclass of information structures with dispositional powers to determine action (both internal and external), and which subsumes desires, goals, intentions, and wishes (Wright, Sloman, and Beaudoin, 1996).

The IAA architecture aims to mimic human behavior and that is why Motivator is an essential component of the architecture. Motivations direct the agent towards its objective. For example, if agent A has performed its job with full motivation and is now waiting for the next job to perform but does not receive any new job then its motivation towards its objective starts decreasing, and if until a specified time there is still no job to perform then the agent goes to sleep mode (this means that the agent would not be using system resources which in turn enhances system response time of other agents). In this example motivator is a function of time. ***Perceptions:*** This module works directly with the Sensors to generate percepts of the agent. A percept is an input from the environment e.g. location of fire and indication of its intensity (Winikoff et al., 2001). Percepts come from the environment through sensors and then translated to generate new events which in turn trigger goals. ***Sensors:*** Sensors of the agent are used to obtain information about the environment through sensing actions. That information is then passed to the perception module to generate percepts of the agent.

5.0 IHK: MULTI-AGENT HEALTHCARE KNOLWEDGE INFOSTRUTURE

After developing the modal and architecture for a single agent that will define the structure of each agent to be used in the IHK multi-agent system, we will, now present the architecture of the IHK (see Figure 3). Here, the architecture specifies the components (agents and modules), their workflow with complete data flow streams and the location of each agent in the system. The IHK architecture consists of five layers: (1) Inter-face layer, (2) Transport layer, (3) Agent application layer, (4) Knowledge description layer, and (5) Object layer.

The Agent application layer is the main processing layer of the IHK as all main agents related with knowledge processing reside on it. The Knowledge description layer deals with knowledge standardization and representation in the IHK. The Object layer defines the multi-modal knowledge for IHK as a HEM (Healthcare Enterprise Memory). The Interface layer deploys the web interface agent as it pro-vides the ability for user to interact with the system. The Transport layer provides a channel for the interface agent and rest of the system to send and receive data and messages.

5.1 Functional Overview of the IHK Infostructure

IHK is a framework consisting of: (a) a HEM (Healthcare Enterprise Memory) which provides the access paths to diverse knowledge sources; and (b) agent-mediated intelligent access to, and retrieval of, heterogeneous knowledge by approximate matching of resources, content navigation, and content correlation. The IHK'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 and adaptation; (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.

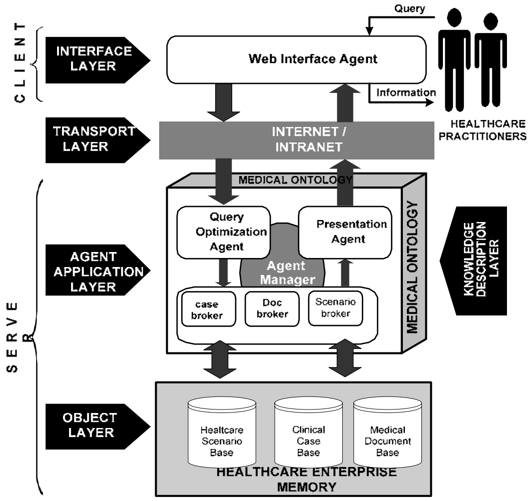

Figure 3 IHK Multi-agent Healthcare Knowledge Info structure

Interaction with the HEM is facilitated by the IHK, 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. Based on such a KS, the manager agent of the IHK will activate the broker agents to retrieve specific knowledge from the constituent repositories—i.e. the case-base, scenario-base and the document-base. Knowledge retrieved from the different sources will provide different perspectives to the user. For instance, the case-base may provide experiential diagnostic-support, the scenario-base may reflect on the tacit knowledge of domain experts and the document base may provide texts, notes, etc. The resulting knowledge package is therefore expected to be 'holistic', and more attractively, it has been retrieved by a simple generic query to the HEM.

6.0 CONCLUSION

We argue that healthcare at all level is a wide open and complex domain associated with complex distributed nature problems space. Intelligent agent technology could provide state of the art solutions for health care problems in multi-agent setting but there is a need for an intelligent autonomous single agent model and architecture mimicking human behavior to leverage the potential of multi-agent technology, in particular health-care domain. We presented in this paper Hybrid Intelligent Autonomous Agent (IAA) Model, Hybrid Intelligent Autonomous Agent (IAA) Architecture, and a Multi-agent Healthcare Knowledge Info structure (IHK).

REFERENCES

Newell, A. (1982). The knowledge level. *In Artificial Intelligence, 18,* 87-127.

Beaudoin, L. (1994). *Goal Processing in Autonomous Agents.* PhD Thesis, School of Computer Science, University of Birmingham.

Maes, P. (1994). Agents that reduce work and information overload. *Communication of the ACM, 37(7),* 31-40.

Aarsten, A., Brugali, D., and Vlad, C. (1996). Cooperation Among Autonomous Agents. *In Proceeding of 4th International conference on control automation, Robotics and Vision,* 3-6. TBD, Singapore.

Wright, I. P., Sloman, A., and Beaudoin, L. P. (1996). Towards a Design-Based Analysis of Emotional Episodes. *In Philosophy Psychiatry and Psychology . 3(2),*101-126.

Bradshaw, J. M. (1997). *An Introduction to Software Agents. In Software Agents,* Bradshaw, J.M. (ed.), Menlo Park, Calif., *AAAI Press,* 3-46.

Chandrasekaran, B., Josephson, J. R, and Benjamins, V. R. (1999). What are ontologies, and why do we need them. *IEEE Intelligent Systems, 14*(1), 20-26.

Winikoff, M., Padgham, L., and Harland, J. (2001). Simplifying the Development of the Intelligent Agents. In the proceedings of the 14th Australian Joint *Conference on Artificial Intelligence (AI' 01) Adelaide.*

John, L. N. and Moreno, A. (2003). Applications of Software Agent Technology in the Health Care domain. *In Whitestein series in Software Agent Technologies.*