MULTI AGENT SYSTEM: AN INFORMATION GATHERING APPROACH

Najeeb A. Al-Samarraie

Abstract

With the availability of vast number of information resources today, a critical problem is how to locate, retrieve, gather, and process information. The concept of an agent has become important in both artificial intelligence (AI) and main-stream computer science. A more modular approach is to build specialized information agents where each agent provides access to a subset of these resources and can serve as one information source to other agents.

This paper surveys agent-based systems information gathering with its problems also describes the term of Multi-Agent systems with its taxonomy, architecture, characteristics, and applications.

Keywords: Agent, Multi-Agent, Distributed Problem Solving, Information Gathering.

1. Introduction

Over the past few decades, the term agent has been applied to many different types of s/w programs sometimes it seem as if any s/w program may be called an agent. An agent is "a software thing that knows no to do things you could probably do your self if you had the time". [1]
Information is useless if it could rot no an easily available when required. As recent years have seen an explosion in the amount of information in electronic from dispersed all over the world, the task of finding the desired information becomes much more complex.

Information gathering (IG) involves searching distributed possibly heterogeneous data source to answer a query. No one source contains all of the information necessary to answer the query. So the information must be gathered and integrated in order to respond Query as if all of the information comes from one source. Agent technology is an obvious candidate to fulfill the demand.
The breakdown of this paper is as follows:
Section 1 notes characteristics, classification, and compare between single and Multi-Agent systems, section 2 describe the ideas of Multi-Agent approach with its reasons for implemented, taxonomy, and applications, section 3 survey the agent-based system information gathering with its problems, architectures, and languages between agents, section 4 concludes the paper.

1-1 characteristics of agents:

The academic communities have been an attempt to define all of the qualities that should make up an agent: [2]

- Delegation: An agent should be able to have a task delegated to it (from a human or other agent) without having all of the details specified.
- Data-directed execution: An agent should be able to monitor its environment to relevant changes and act upon those changes.
- Communication: An agent should be able to communicate with others (humans or agents) to accomplish a task.
- Reasoning: An agent should be able to reason about the arguments with which it is invoked.
- Planning: An agent should be able to formulate "plan to utilize available resource (including other agents) to accomplish a task".
- Autonomy: The agent operates without direct intervention.
- Actuation: The agent needs to be able to affect its environment via an actuation mechanism for autonomous operation.
- Intelligence: The agent needs to be able to interpret the monitored events to make appropriate actuation decisions for autonomous operation. In addition the basic attributes mentioned above, an agent may have other attributes such as mobility, security and other, high level attributes of an agent model.

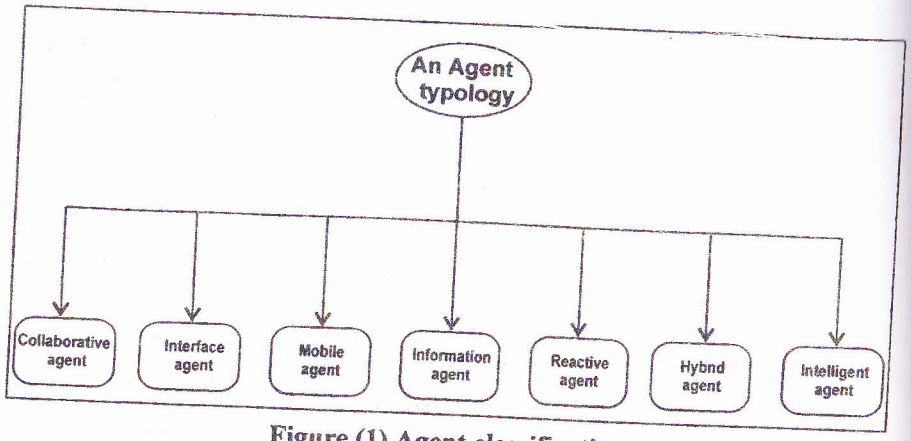

Figure (1) Agent classification

1-2 Classification of Agents: [1]

There is no easy and straightforward way of grouping agents into certain classes. There are some properties which make the job easy.

First: Whether the agent capable of moving around the network, or is the agent bound to a some specific location. (Static) or (mobile) agents.

Second: Whether the agent responds on the environment changes (reactive agents), or does posses some sort of a deliberate behavior (Deliberative agents).

Third: According to (British telecommunication) several ideal, and primary attributes, which an agent should exhibit, classifies B.T labs. The mineral characteristics are autonomy, co-operation and learning. An agent classification shown in Fig.(1).

a. Collaborative Agents:

It consists of the following characteristic autonomy, social ability responsiveness and proactiviness.

b. Interface Agents:
Interface agent's emphasis autonomy and learning with other agents in order to complete task given by their user.
Interaction between human and interface agent does not necessary required an agent communication languages as the communication between agents does interface agents typically learn as they quid's their users. The ways how agent can learn form their users:
- By observing and imitating.
- Receiving positive and negative feed back.
- Receiving clear instructions.
- Asking advice from other agents.

c. Mobile Agents:
Mobile agents are computational S/W processor capable of migrating from one computer to another via the Internet or Intranet.
Th security issues will play a great role when designers are dealing with mobile agents for example, viruses have the same behavior agent do this yields the problem how to exclude viruses from agents.

d. Information Agents:
It is a tool for gathering constantly growing information from the Internet information agents perform the role of managing, manipulating and collating the retrieved and

distributed information. Information agents can have many different roles depending on their job in current application.

e. Reactive Agents:
In is belong to a special category because they do not posses any kind of symbolic model of their environments. Therefor they act respond in a stimulus-response manner to the present state of the environment in which they are embedded.

f. Hybrid Agents:
Hybrid agents are combination of agents mentioned above since each type has its own strengths and deficiencies, the aim is to find the best suitable solution for the problem in hand, where deficiencies are minimized and the strengths are maximized in a straight forward manner.

g. Intelligent Agents:
Intelligent agents always have some sort of a decision-making model, which will give the agent a primitive level of intelligence. Basically an agent is intelligent, if its perceives its environment and capable of reasoning its perceptions, solve problem, and determined actions depending on its environment and tasks. [3]

1-3 Single Agent vs. Multi-Agent

Single agent system is based on the centralized process model, where multi-agent systems on the contrary are distributed systems. Centralized a single agent, which makes all the decisions while the other act as remote slave. In a single system the agent models itself as part of the environment, which has some extra-environmental components such as actuators, sensors and evens other agent. However other agents are considered to behave like an actuator or sensor in sense that they do not possess any own goals Fig. (2)

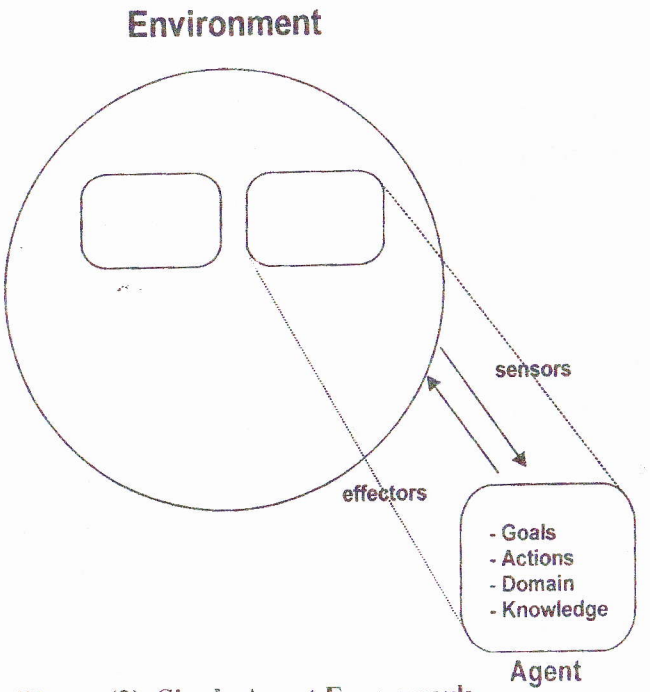

Figure (2) Single Agent Framework

Multi-agent system differ from single-agent systems in that several agents exist and they are aware of each others goals and actions. In addition to that the agents can be engaged in conversation with each other.

From the individual agents point of view the most significant difference is that other agents can determine the environments dynamics. Simply the amount of available resources varies depending on the state of other agents. [4]

2-Multi-Agent System

The term Multi-Agent systems are currently in vogue, and have been generally applied to any system that is , or can be considered to be, composed of multiple interacting agents. In the various multi-agent (or, more properly, Multi-Agent) system that have been proposed or developed, a wide variety of "agent" have been considered, ranging from fully autonomous intelligent agents (such as people) down to relatively simple entities (such as rules or clusters of rules).

Multi-Agent System (MAS) is a loosely coupled network of problem-solver entities that work together to find answers to problem that is beyond the individual capability knowledge of each entity Fig. (3)

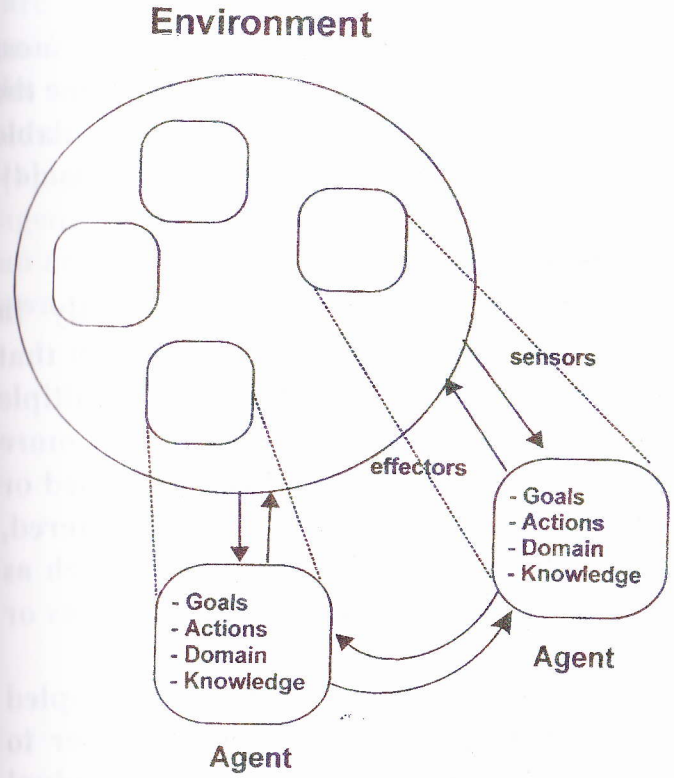

Figure (3) Multi classification

general the term Multi-Agent system covers all types of stems composed with multiple autonomous component th the following characteristics: [1]

ach agent has incomplete capabilities to solve problem.

here is no global system control.

ata is decentralized.

ommutation is asynchronous.

2-1 Why Multi-Agents

The main reasons for implementing a Multi-Agent system are:

a. Some domain required it.
b. Parallelism.
c. Scalability.
d. Simpler programming.
e. to study intelligence.

2-2 Taxonomy

More recently, when Multi-Agent Systems (MAS) have become a popular research field there have been introduced a new taxonomy from MAS perspective. Their characteristics are:

- System function.
- Agent architecture (degree of heterogeneity).
- System architecture (communication, protocols).

2-2-1 Homogenous Non-Communicating MAS

It is the simplest MA involves homogenous non-communicating agents. In this system:

- all agents, have the same internal structure including goals, domain knowledge, and possible actions.
- they have the same procedure for selecting among their action.

2-2-2 Heterogeneous Non-Communicating MAS

Adding the possibility of heterogeneous agent in MAS domain adds a great deal of potential power to the system.

An important sub dimension of the heterogeneous agent system is whether agents are benevolent or competitive.

2-2-3 Heterogeneous Communicating MAS

The full power of MAS can be realized when adding the ability for agents to communicate with one another.

2-3 Co-Ordination in Multi-Agent Systems

Co-Ordination is the central issue in MAS in practical and in Distributed Artificial Intelligence (DAI) in general.

To achieving coherence in the MAS the need for Co-Ordination is absolute. Coherence in context means that agent actions goes well, and that they do not conflict with one another, in other words, coherence refer to how well a system of agents behaves as a unit. The reasons why MAS needs Co-Ordination are:

- Preventing anarchy or chaos.
- Meeting global constraints.
- Distributed expertise.
- Dependencies between agents actions.
- Efficiency.

Several approaches have been introduced to a chieve Co-Ordination in MAS:

- Organizational structuring.
- Contracting.
- Multi-agent planning.
- Negotiation techniques.

2-4 MAS and Distributed Problem Solving (DPS)

Distributed Artificial Intelligence (DAI) has existed a sub field of AI for less than two decades. DAI is concerned with systems that consist of multiple independent entities that interact in domain. Traditionally, DAI has been divided into two sub-disciplines, Distributed Problem Solving (DPS) focus on the information management aspect of systems with several component working together towards a common goal; Multi-Agent systems (MAS) deals with behavior management in collections of several independent entities, or agents.

There are three views of the relationship between DPS and MAS. They are not mutually exclusive and in fact build upon each other to some extend. [5]

View1 is DPS is a subset of MAS when certain assumptions hold, including the benevolence assumption, the common goals assumption. This view focuses who made the system (by single designer or designers with shared goals?).

View2 MAS provides a substrate for DPS. This view focuses on how the system was made (were individuals thrown together or were team interactions relied upon?).

View3 MAS and DPS are complementary research agents. This view focuses on why the system was mad (was it mad to ask questions about the impact of changing environment or of changing agent population?

2-5 MAS applications

The main applications of MAS are:

- Electronic commences.
- Network monitoring and control.
- Transportation systems modeling a control.
- Information systems.
- Automated meeting scheduling.
- Industrial manufacturing and production.
- Electronic entertainment.

3. Agent for Information Gathering

With the proliferation of electronically available information, information is becoming increasingly more difficult for a person or machine system to collect, filter, evaluate and use in problem solving. As a result, the problem of locating information sources, accessing, filtering, and integrating information in support of decision making, as well as coordinating information retrieval and problem solving efforts of information sources and decision making systems has become a very critical task.

Most current agent-oriented approach have focused on what we call Interface agent, a single agent with simple knowledge and problem solving capabilities whose main task is information filtering to alleviate the user's cognitive overload.

3-1 Limitations of singe agent

There are some limitations for centralized approach:

- Single general agent would need an enormous amount of knowledge to be able to deal effectively with user information request that cover a variety of tasks.
- Centralize system constitutes a processing bottleneck and "singe point of failure".
- Unless the agent has beyond the state of art learning capabilities, it would need considerable reprogramming to deal with the appearance of new agents and information sources in the environment.
- 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.

New proposed solutions to use multi-agent computer systems to access, filter, evaluate and integrate this information. The distributed architecture has been motivated by the following considerations: [6]

- Distributed information sources.
- Sharability.
- Complexity hiding.
- Modularity and reusability.
- Flexibility.
- Robustness.
- Quality of information.
- Legacy data.

The above considerations clearly motive the development of systems of distributed s/w agents for

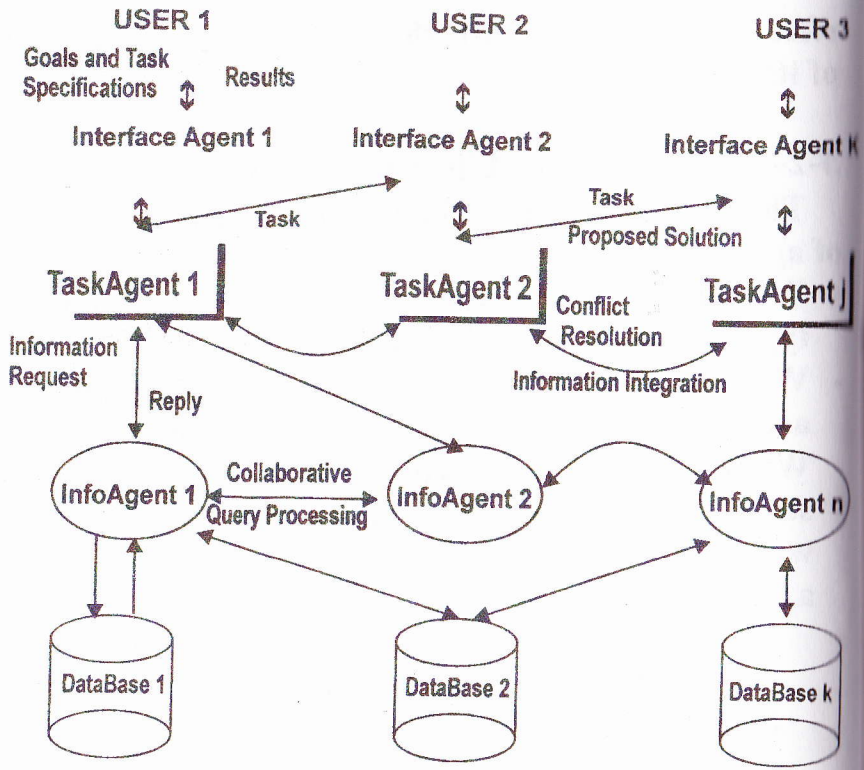

Figure (4) Distribution Agent - Based Architecture

Information Agent(s) provide intelligent access to a heterogeneous collection of information in response to queries. Information assistants have models of the associated information resources, and strategies for source selection, information access, conflict resolution, and information fusion. For example, an agent that monitors stock prices an information agent.

The architecture of all the above agents follows the general BDI type philosophy, however, each of them embodies particular architectural design commitment, to make them

information gathering and decision support in the Internet-based information environment.

3-2 Agent Architecture

There have been several proposals for generic agent frameworks. The distributed agent-based architecture has three types of agents; Interface Agents, Task Agents, and Information Agents.

Interface Agent(s) interact with user receiving user specifications and delivering result. Its acquire, model, and utilize user preferences to guide system coordination in support of the user's tasks. For example, an agent that filters electronic mail according to its user's preference.

Task Agent(s) support decision making by formulating problem solving planned carrying out these plans through querying and exchanging information with other software agents. They have knowledge of the task domain, and which other task assistant or information assistants are relevant tot performing various parts of task. For example an agent that makes stock buy or sell recommendations is a task agent Fig. (4)

effective in dealing with the particulars category of issues of its type. [7]

3-2-1 Architecture Characteristics

The crucial factors influencing the determination of type of agent are:

- What is the functional and informational scope of these types of agents in a distributed architecture?
- What kinds of interactions, coordination and actuation are predominant among these agents, and
- What reusable agent components can constitute agent structuring and what functionality are these components will support:

a. Agent Functionality:

The main function of an interface agent include:

- Collecting relevant information from the user to initiate a task.
- Presenting relevant information including results and explanations.
- Asking the user for additional information during problem solving.

A task agent perform most of the autonomous problem solving:

- Receives user delegated task specifications from an interface agent.
- Interprets the specifications and extracts problem solving goals.
- Forms plans to satisfy these goals.
- Identifies information seeking sub-goal that are present in its plans.

- Decomposes the plans and coordinates with appropriate task agents or information agents for plan execution, monitoring and results composition.
- An information-specific agents primarily provides intelligent information services and retrieval of information in response to query.

b. Agent Knowledge

An interface agent has the following knowledge:

- A module of the user's goals and preference pertaining to a task.
- Knowledge of the relevant task assistant that can perform the task.
- Knowledge of what must be displayed to the user and in what way.
- Protocols for interacting with relevant task assistants.

Task agent has the following knowledge.

- Model of the task domain.
- Knowledge for performing the task.
- Information gathering needs associated with the task model.
- Knowledge about relevant task or information-specific agents that it must coordinate with support of its particular task.
- Protocols that enable coordination with the other relevant agent.
- Strategies for conflict resolution and information fusion.

A typical information-specific agent knows.

- Model and associated meta-level information of the databases that it is associated with such as size,

average time it takes to answer a query and monitoring cost of query processing.
- Procedure for accessing databases.
- Conflict resolution and information fusion strategies, and
- Protocol for coordination with other relevant software agents.

c. Agent Organization

The system organization has the following characteristics:
- There are a finite number of task assistants that each agent communicates with.
- The information assistant is responsible for recognizing important information, information filtering, and checking information quality. Task assistants are responsible for resolving information conflict and integrating information from heterogeneous information sources for their respective tasks.
- The task assistants are responsible for activating relevant information assistant and coordinating the information finding the filtering activity for their task.

3-3 Problem of Information Gathering (IG):

IG involves searching distributed, possibly heterogeneous data sources to answer a query, one source contains all of the information necessary to answer the query, so the information must be gathered and integrated

in order to respond to a query as if all of information come from one source.

In simple example the user must know what information is available, how it is distributed about the various information sources, what order to access the different data sources, and how to access each different type of source. An automated information gathering system would allow the user to pose a query with having to know how the information is distributed among different sources. [6]

3-3-1 What agents can do for information gathering:

Looking again at the quality of an agent, we see that each quality can benefit the information gathering task, some examples.
- Delegation- An agent can receive a query while the data sources necessary to answer the query are not specified. Recall that one of our goals of Information Gathering is transparency the user should not need to specify how data is distributed about the information sources.
- Data directed execution- Agent can notify users of updates in information sources or revised pattern.
- Communication- Agents can communicate with other agents to obtain information from resources to which they do not have access.
- Reasoning- An agent can optimizes a given query to minisize communication, time and expense.
- Planning- An agent can formulate a plan to access appropriate resource to answer a query.

There are two other quite useful qualities in the (IG), Mobility and distribution of agents.

3-4 Agent Communication Languages (ACLs)

Agent-oriented software engineering must facilitate the interoperation of heterogeneous agents.

As standard Agent Communication Language (ACL) the prime requirement for interoperability. [2,6]

The requirements for an effective communication standard are:

1. Form: The language should have good form concise but readable and extensible.
2. Content: Distraction should be made between the message continue test and well-defined set of communicative acts.
3. Semantics: Semantics required formal description of the meaning of communications primitives, addressing concepts of location and time.
4. Implementation: The language implementation should have a simple interface which hides implementation details, allows a subset of the primitives to be used and is efficient.
5. Networking: Networking requirement includes support for synchronous, asynchronous multicast, broadcast, and point-to-point messaging over various transport protocols TCP/IP (email, http,etc).
6. Environment: The language should be operable in distributed, heterogeneous, and dynamic environments, interoperable with other languages and protocols and adaptable to legacy systems.

7. Reliability: The language should ensure reliability, robustness and security.

There are number of ACL, KQML (Knowledge Query and Manipulation Language) KIF (Knowledge Interchange Format).

Coordination languages, in contrast to ACLS, which transfer only single messages, coordination languages encapsulate entire conversation. [8]

2-5 Architectures

Most architecture in multi agent IG systems is very new and their numbers are still growing at an accelerated rate. [2]

Most architecture is quite similar on the surface but have subtle difference when worked at move closely.

Several of architectures are mediators, match marker, facilitator and broker.

3-5-1 Mediator

A mediator provides a transparent interface to heterogeneous data sources. It decomposes queries translates sub-queries into schemas understandable by information sources, distributes the sub queries and integrates the responses. Fig. (5)

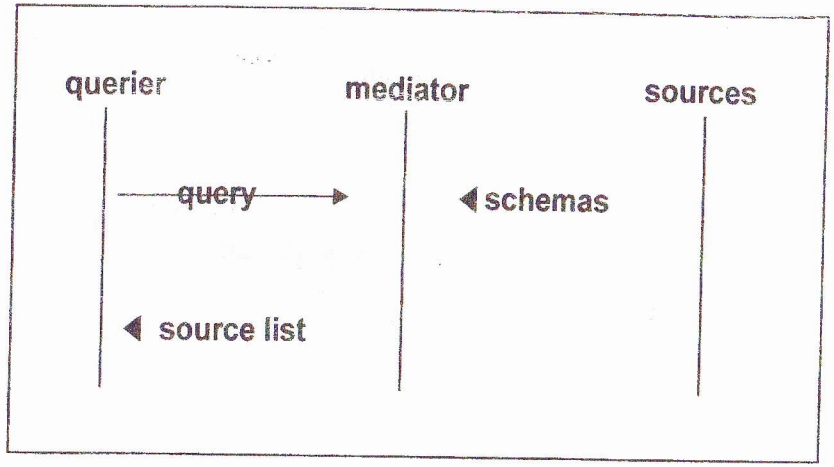

Figure (6) Matchmaker Pattern Diagam

While the mediator is fully involved in query decomposition and translation. In match marker scenario these responsibilities are created separately. The major benefit of using a match marker agent is that it provides a central location for global resources. A major disadvantages of using this architect are that it can create a resources. A major disadvantages of using this architect are that it can create a bottleneck if many agents must make frequent use of the match marker.

3-5-3 Broker:

A broker agent maintains a central list of service providers, as does a match marker agent.

The broker: is able to reduce the communication overhead, it is no longer necessary to send a list back to the querier. Fig. (7)

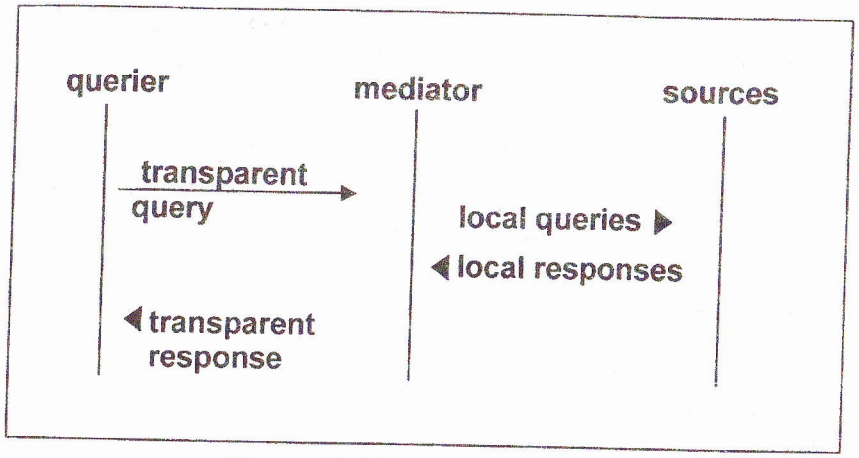

Figure (5) Mediator Pattern Diagram

The main advantage to this hierarchical mediator architecture is the uniformity across the system.

3-5-2 Match Marker

Given a query, a match marker agent chooses the source(s) which is best to answer the query by using a central list of sources, which it maintains.

The match marker has no further business with query once the appropriate sources are found see Fig. (6)

The advantages of the facilitator over both the match marker and the broker agents is that it embodies not only a central resource capabilities model but also a central schema. Fig. (8)

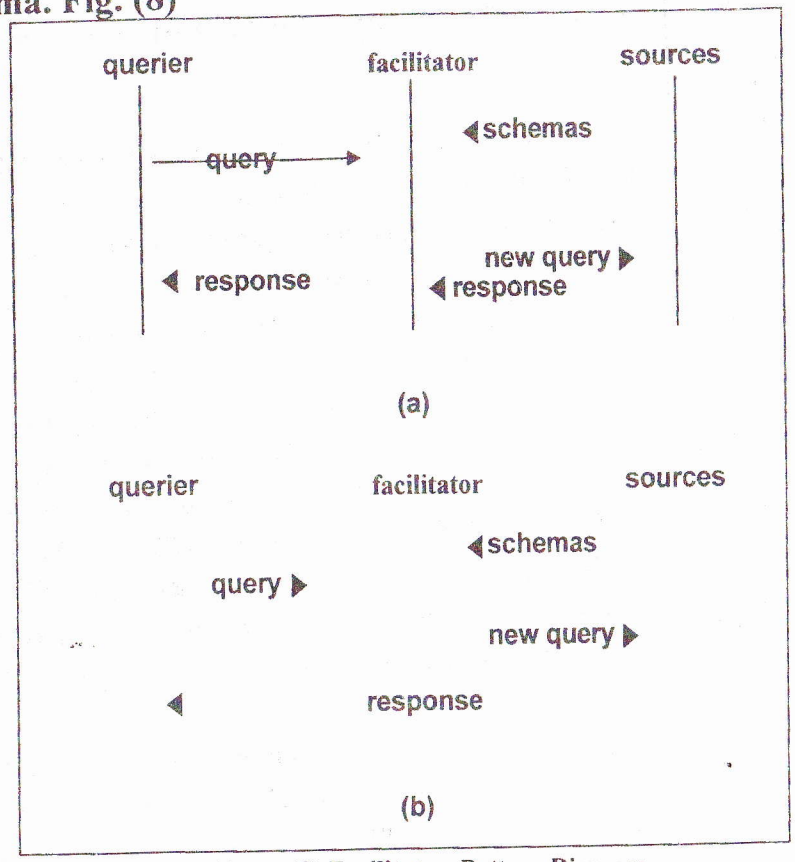

Figure (8) Facilitator Pattern Diagram
(a) Response through the Facilitator
(b) Response Directly

Finally the above architectures the mediator performs the most tasks and requires the least inter-agent communication while match marker performs only one of those tasks and forces.

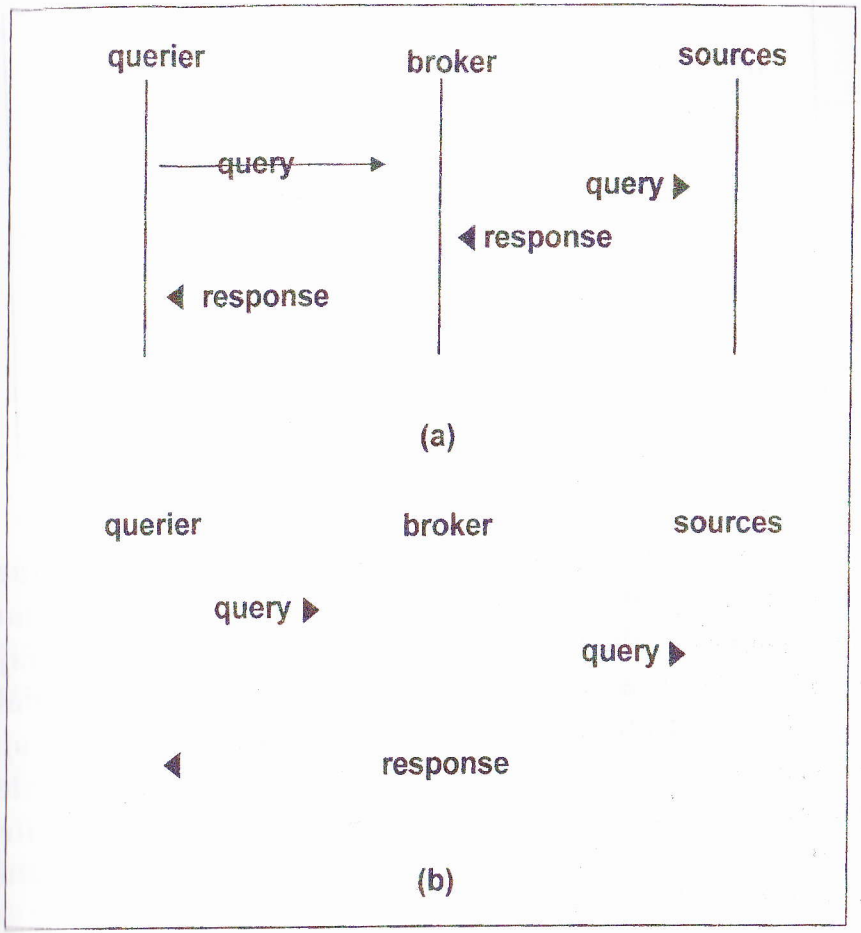

Figure (7) Broker Pattern Diagram
(a) Response through the Broker
(b) Response Directly

The broker suffers from the same disadvantages as the match marker.

3-5-4 Facilitator

It is similar to broker, the difference is in the schema translation responsibilities.

More communication between agents to accomplish the remaining tasks. Correspondingly, a match marker is the easiest type of IG agent to update or change, while the mediator is a much more complex entity and is the most difficult to maintain.

4 Conclusion

This paper has reviewed the main concept and issues associated with Multi-Agent Systems (MAS) and Information Gathering Systems (IG). Information gathering agent problems has recently received considerable attention due to the growing number of structured information sources available on line. The ability of arrange specialized agents in hierarchical manner can allow distributed query planning, with agents at lower levels planning more detailed tasks. This is the idea behind the Multi-Agent Planning Architecture (MPA). One problem with these planning schemas is the possibility for bottlenecks in communication. This is an area in which work still remains, and can be beneficial for complex IG tasks.

There is ongoing research in applying IG agent systems to various problems and integrating them with other system, also research include coordination of multiple mobile agents for IG, and security issues for mobile agents.

References

1. H. Lothman. Intelligent Agent Platform in Distributed Software Architecture Ms.c thesis Degree

programming in Knowledge Engineering, OULU Polytechnic, 2001.
2. C. Carrick, S. Hayden, and Q. Yang. Agent Systems for Information Gathering The Knowledge Engineering Review 11(3), 1996.
3. M. Wooldridge, and N. R. Jennings. Intelligent Agents: Theory and Practice Proceeding of the Thirteenth International Jount Conference on AI 1994.
4. K. Sycara, and D. Zeng. Coordination of Multiple Intelligent Software Agents. International Journal of Cooperative Information Systems volume(2), 1995.
5. E. H. Durfee, and J. S. Rosenschein. Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples.
   The Knowledge Engineering Review 10(2), 1995.
6. C. A. Knoblock, and J. L. Ambit. Agent for Information Gathering.
   IEEE Expert October 1997.
7. K. Sycara, and D. Zeng. Multi-Agent Integration of Information Gathering and Decision Support.
   12th European Conference on Artificial Intelligence 1996.
8. T. Oates, M. V. Nagendra, and V. R. Lesser. Network Information Retrieval as Distributed Problem Solving.
   Umass Computer Science Technical Report August 1998.