



Multi-Agent Systems: A Review Study

Hanan Qassim Jaleel

Baghdad College of
Medical Sciences,
Baghdad, Iraq

hancollege2000@gmail.com

Jane Jaleel Stephan

University of Information
Technology and
Communications,
Baghdad, Iraq

janejaleel@uoitc.edu.iq

Sinan A. Naji

University of Information
Technology and
Communications,
Baghdad, Iraq

dr.sinannaji@uoitc.edu.iq

Article history: Received 21 April 2020, Accepted 17 June 2020, Published in July 2020.

Doi: 10.30526/33.3.2483

Abstract

Multi-agent systems are subfield of Artificial Intelligence that has experienced rapid growth because of its flexibility and intelligence in order to solve distributed problems. Multi-agent systems (MAS) have got interest from various researchers in different disciplines for solving sophisticated problems by dividing them into smaller tasks. These tasks can be assigned to agents as autonomous entities with their private database, which act on their environment, perceive, process, retain and recall by using multiple inputs. MAS can be defined as a network of individual agents that share knowledge and communicate with each other in order to solve a problem that is beyond the scope of a single agent. It is imperative to understand the characteristics of the individual agent or computing entity to distinguish a simple distributed system and multi-agent system. To solve each task, the agent can take the appropriate actions, decisions, and can cooperate with other agents to fulfill its goals or objectives. MAS has several applications including cloud computing, web advertisement, e-commerce, computer network, learning abilities, etc. This survey represents a comprehensive discussion of all aspects of MAS, and discusses an overview of MAS definitions, architecture, features, communications, organizations, and finally discusses the some of the real practical applications.

Keywords: Agent organizations, MAS applications, MAS architecture, MAS communications.

1. Introduction

Multi-Agent Systems (MAS) are systems that consist of a collection of interacting computing elements known as agents [1]. These agents act as complete autonomous entities that can interact with each other and decide about what and how to perform different kinds of tasks to achieve their own and the other goals in an environment [1, 2]. For example, if there are different parties



(e.g., organizations) with multiple goals and have propriety data and domain knowledge, then a multi-agent system is the best way to model their interactions even though some parties wish to build the systems that represent and serves them [2, 3]. On the other hand, it is impractical to build a single system that represents all of them. Generally, organizations do not get ready to give their data and do not want to get their data controlled by the representative of another organization. So, Multi-Agent systems can be considered as a network of agents that share knowledge and cooperate with each other[4]. Furthermore, many MAS models are designed in such a way that the principal problem is divided into several manageable sub-problems that can be further divided into different problems solving agents with their own interests and goals. Generally, multi-agent systems come in the category of relatively new sub-field of computer science which is based originally on the concepts of artificial intelligence and distributed systems. The field has gained widespread attention since the 1980s and witnesses prompt progress motivated by the premise that multi-agents are considered as an appropriate model to exploit the possibilities presented by massively distributed artificial intelligence systems. Wooldridge [5]. Stated that “Multi-agent systems seem to be a natural metaphor for understanding and building a wide range of what we might crudely call artificial social systems”. The main concepts of Multi-agent systems are not limited to a specific domain and they are found in different application domains[6].

In the work of Balaji and Srinivasan, [1]. They presents the agent and MAS properties and focuses on organization types, decision making methods, as well as coordination and learning methods. Dorri et al. [2]. Concentrate on MAS challenges and discuss some MAS applications in networks, robotics, modelling complex systems, city and built environments and smart grids. Glavic, [3]. Emphasizes on MAS defenitions and concepts specifically to the power engineers concerning these methodologies in order to solve complex real problems, with providing examples in order to establish load shedding systems. Chrysanthi et al. [4]. Considers agent communication languages (ACL), the ontologies and frameworks used to construct MAS, and the role of agents in virtual enterprises. In the work of Mostafa et al.[7], the milestone achievements of the agent technology conceptualization, modeling and improvement platforms have been reviewed and defined. In the prior work of Abbas et al., [8]. They focus on presenting MAS organizations encompassing prototypes and methods used for arranging agents in a static or dynamic way in MAS. Chin et al., [9]. Concentrates on different types of agent architectures and their properties in order to use them to resolve practical problems.

This paper aims to discuss the main issues of multi-agent systems, their properties, types, organizations, architectures, communication methods and agent communication languages. Moreover, it reviews recent applications that are considered an important part of the literature to the new researchers in this field, such as cloud computing, image segmentation, computer networks, e-commerce, and web advertisements. It compares between two proposed research techniques for each application, with their contributions, benefits, architecture and limitations.

The rest of the paper is arranged as follows: Section 2 provides an introduction to agents and their characteristics. Section 3 presents the agent environment along with its features. Section 4 provides an introduction to MAS and the main differences between MAS and other similar systems. Sections 5,6,7,8,9,10 and 11 discuss MAS organizations, MAS architecture, Agent types, MAS communication, Agent Communication Languages (ACL), Ontology and

JADE (Java Agent Development Environment) respectively. The final section 12 presents the conclusion.

2. Agent Definition and Characteristics

The term “agent” has been widely used in many technologies and fields such as artificial intelligence, distributed database systems, cloud computing, operating systems, computer networks, etc. Since the agents can be found in many forms and in many application fields which are varied significantly, researchers do not agree on a uniformed definition of "Agent"[1, 4]. Most definitions agreed upon defining agent as followings: autonomy entities which are able to perceive input from its environment through sensors and parameters, act upon that environment through effectors in order to fulfill the desired goals. The agent can be viewed as a software component that has flexibility and autonomy and gives an interoperable interface to an arbitrary system. The agent system can be constituted from a single agent working in its environment and can interact with its users, or can consist of multiple cooperative agents which can model sophisticated systems and present the possibility of agents that have shared their goals. By acting in their environments, agents can interact and cooperate with each other indirectly, or they can use communication language (ACL) to act directly[1, 5].

Autonomy in agents means that the computational agents can work without the intervention of other entities and have control over their activities. Giving the goals to agent implies that the agents should perform actions in the environment in order to fulfill their determined goals. Behavior means that agents can perform actions after receiving sensory inputs from the environment [3]. A framework that depicts a single agent is presented in **Figure 1**.

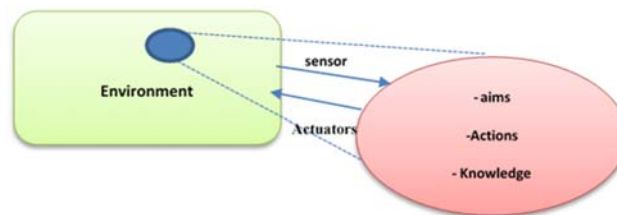


Figure 1. The agent structure [3].

Besides sensory inputs, actions and goals, the agent can have knowledge (which is about a specified environment or a problematic domain). This knowledge may be based on artificial intelligence (such as rule-based, machine learning, etc.)[5]. Different definitions of agents include several properties which may be summarized briefly in **Figure 2**. As follows:

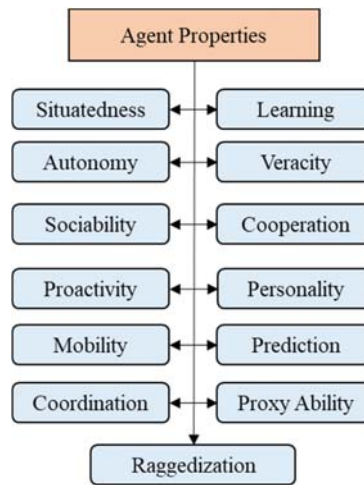


Figure 2. Agent properties.

1. Situatedness

The agent can interact with its environment by using sensors and perform actions using their actuators. All the sensory inputs are received directly as a result of agent interactions in their environment. The agent acts immediately with its environment and does not act as a level advisor. This differentiates the agent from expert systems where the suggestion for decision making is performed through a middle (broker) agent and does not impact directly on the environment[3, 6].

2. Autonomy

This means that the agent can act and choose its actions independently without interventions by other agents or human. This feature protects the internal states of the agent from extremal impact and gives the agent control over its own activities[2, 6].

3. Sociability

In reaching their goals, agents share and collaborate their knowledge, and demand information from other agents to enhance their performance. An agent can be considered to be social because it participates with humans or agents to perform its goals[4, 7].

4. Proactivity

Agents can use their sensed parameters, Data and information from other agents to predict or expect future actions. These predictions permit agents to perform actions that satisfy their goals. This means that the same agent may act using different actions when suited in various environments. An agent is said to be proactive, since it does not behave in its environment but is capable of presenting goal- Directed behavior[2, 10].

5. Reactivity

An agent is considered to be reactive because it can receive a sensory input from its environment, and responds to changes that happen in its environment in some such way[2, 4].

6. Mobility

An Agent can transfer from one computer to another and through various architectures and platforms[1, 4, 6].

7. Coordination

An agent is capable to achieve activities in a common environment with other agents. Activities are regulated by means of plans, roadmap or other process management techniques[4].

8. Learning and Adaptivity

An agent is able to respond to other agents and to its environment. More improved kinds of adaptation allow an agent to change its behavior using its experience. An agent is able to learn from its experience as well as learning from the environment and cooperation with other agents[2, 4].

9. Veracity

An agent may be truthful, giving the assurance that it will not intentionally transfer (or communicate) false information [4, 7, 10].

10. Cooperation or Collaboration

The agent is capable of cooperating with other agents to achieve a given purpose. Non-combative agents that win or fail together [4, 6, 10].

11. Personality

An agent has a well-known rational character[4, 6].

12. Prediction

An agent is said to be predictive if its anticipation of how the environment works is accurate enough to permit it to predict correctly how to achieve a given goal[2, 6].

13. Proxy Ability

An agent can perform on the side of a user or can be a representative for the assistance (or utility) of other entities [4, 11].

14. Raggedization

An agent is capable of dealing with faults and exhibits robustness in behavior against incomplete and inconsistent data[4]. **Figure 3.** Depicts the sample blocks of an agent.

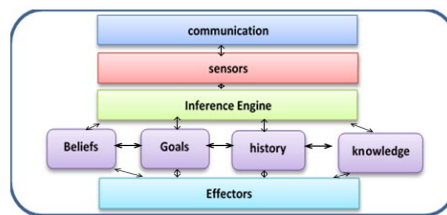


Figure 3.The agent sample blocks [1].

3. ENVIRONMENT

The environment refers to the place where the agent is located. It impacts agent behavior and can also be influenced (or impacted) by the agent. The environment encompasses the agent and it is not a portion of the agent. The environment is different from the agent and its actions. The agent can use sensory input from the environment for making decisions. An agent can seek and explore the environment in the same way a human user does in the real environment.

The behavior of an agent not only changes its environment but also affects the neighboring agents. In order to choose the best action, the agent should predict the behavior of other agents in the environment. This kind of simultaneous learning can cause weak behavior. When the environment is dynamic, the situation is more complex. The agent has to distinguish among the

actions of other agents and the changes in the environment[3, 6, 12]. The environment enjoys some properties that we can list and illustrate in **Figure 4**.

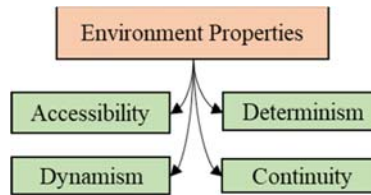


Figure 4. The Environment Properties.

a) Accessibility

Accessibility means the ability of agents to sense sensory data from the environment. An environment can be considered accessible if the agents sense precise and updated information from it. An inaccessible environment means that the agent cannot sense complete data; it may sense noisy or perturbed data from it. Physical-world and any real-world environment can be considered as the inaccessible environment in this way[1, 2, 12].

b) Determinism

A deterministic environment is one who’s each action has a unique assured effect; there is no doubt or confusion about the situation that will be resulted from achieving the action. In a deterministic environment, the outcomes can be anticipated. For example; any environment that can be (designed or simulated) by the finite state machine, the agent accurately realizes the next state of behavior. In a non-deterministic environment, the results of performing an action are not totally expected. Results can be affected by other circumstances. For example, by playing a game, the result may depend on the behaviors of all players[1, 2, 6].

c) Dynamism

A static environment is proposed to be still unchanged, except by the actions and behaviors by agents. The static environment has changes that may happen only as a result of agents’ behaviors; otherwise, it might be dynamic. The real world can be considered as dynamic environment[2].The decision making depends on the sensed input data from the environment. The sensed data may not be precise when the environment varies. Therefore, in a dynamic environment, the agents should identify changes in the environmental situation and perform updating the sensed data which has acquired computation overhead in comparing with static environment[12].

d) Continuity

The environment can be divided into two types: discrete and continuous. An environment is considered to be discrete if these are defined, limited and finite numbers of behaviors and perceptions in it [12]. The discrete environment restricts an agent to a set of predefined situations. On the other hand, a continuous environment influences the agent's situation via a continuous function. A chess game can be considered as an example of a discrete environment, while taxi driving is an example of a continuous environment[13].

4. Multi-Agent System (MAS)

A Multi-Agent System (MAS) is such an advanced technology that a number of autonomous agents perform actions in the environment and achieve the desired goal. The

agents either cooperate or compete, either they share or not the common pieces of information with other agents. The considerable properties of MAS involving flexibility, reliability, efficiency, make it an efficient methodology to solve sophisticated problems [1, 12]. MAS is effective since it divides the tasks into multiple smaller subtasks, each of which is given to a determined agent. The processing overheads are distributed among several agents, which results in lowered cost as compared to a method that can resolve a problem using a single agent only. Using its predetermined knowledge, each agent can perform its assigned task with some flexibility. When one agent in MAS fails, the task may be assigned to other agents to be performed[13].

Some of the advantages of utilizing a MAS system are:

- 1- MAS offers increased speed and efficient operation because of the parallel and asynchronous computation.
- 2- A smooth degradation and downgrading of MAS are possible due to the failure of one or more agents; therefore, it increases system robustness, trustworthiness, and reliability.
- 3- Agents are added as needed to the system, which increases system stability and makes it more flexible
- 4- The cost of the system is reduced because independent agents' costs are less than a centralized system.
- 5- Reusability: Agents may have a modular architecture and they are substituted in other systems and can be updated easily than a consistent system.

5. MAS Properties

MAS can have several properties that can be outlined and shown in **Figure 5**. As follows:

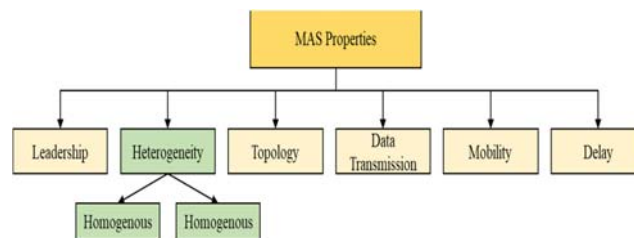


Figure 5. The MAS Properties.

5.1. Leadership

The MAS has a leader i.e. an agent that identifies goals and objectives for other agents depending on one common goal. The MAS can be categorized as leaderless (without a leader) or leader- followers (have a leader and follower agents). In the leaderless system, each agent decides its activities depending on its goals and the decision is influenced by other agents' decisions if agents cooperate to reach an agreement on a certain point. In a leader follower's system, the leader identifies and determines actions for other agents (followers) that communicate to determine the location of the leader agent. The leader agent is selected by other agents and the system can include one or more leader agents. The follower agents may require keeping track of the location of a mobile leader which draws more computational overheads. When there are more than one leader agents, they can cooperatively guide other agents via communicating with other agents in order to perform taking a decision[1, 3].

5.2.Heterogeneity

MAS can be categorized (or classified) into homogenous MAS and heterogeneous MAS, depending on the architecture of autonomous agents that form the system[3].

- **Homogenous structure**

The agents that continue MAS include the same internal structure. Internal structure means that all agents have the same goals, sensory abilities, states, reasoning mechanisms and actions [3]. Agents can be differentiated by their physical positions and the places of performing their actions in the environment. The agent may receive sensory input from various portions of the environment. There may be overlapping in receiving sensory data[13, 14].

- **Heterogeneous Structure**

In this structure, the agents may vary in their capabilities, architectures, and performance. Because of the dynamism of the environment and the position of a certain agent, the actions selected by each agent varies from the agent which is placed in different portions, although it can perform the same function [3,13]. Heterogeneous structure can model and simulate applications which are nearest to real- world problems. Each agent contains different goals and objectives which may contradict with the goal of other agents[8].

5.3.Topology

It means that the position and relationships of an agent. The topology can be divided into a static and dynamic topology. In static topology, the location and relationship among agents are stationary without changes during the agent's lifetime. In dynamic topology agent's location and relations vary while the agent is moving, goes away or joins the system, or establishes new relationships with other agents [3].

5.4.Data transmission

Agent can perceive the environment and collaborate the sensory data with neighboring agents in a time-triggered manner such that the agent can frequently perceive the environment, sense and collect information in a predetermined time. Agents can also transmit all the new perceived sensory information to other agents. In event-triggered system, the agent just perceives the environment when a certain event occurs. Then, the agent sends the sensed data to other agents in MAS[8, 15].

5.5.Mobility

Agents can be categorized into static or mobile agents. A static agent is placed constantly on the same location in the environment while a mobile agent uses the resources of neighboring agents, keeps an eye on them and perceives the environment from the location and place of other agents in order to achieve actions [2, 8].

5.6.Delay

The agents can meet several kinds of delay in performing their actions. In exchanging data, agents may face delay in the medium of communication either communication is wired or wireless. MAS can be divided into two types with delay system and without delay. The latter is simple since there are no delays in communication and computation. Although most of the physical real environments present several delays[2, 15].

6. Distinguishing MAS with Expert System, and Object-Oriented Programming

These are three differences that can be outlined between the agent and object view as follows[2, 5].

1. The agent may incorporate a stronger concept and idea of autonomy compared to objects, and especially it determines whether or not achieve an action on demand from other agents in the system.
2. Agents have flexibility in these actions, allowing them to be more reactive and social in the environment, while the object model does not have these types of behaviors.
3. MAS is naturally a multi-threaded system such that every agent is proposed to control one thread somehow[2, 5, 12].

An expert system is able to resolve problems and offer pieces of advice in the domain of knowledge to solve the problem. The main distinctions among agents and expert systems are:

1. Expert systems do not interact and are not coupled with environments. They do not use sensors to obtain their information; instead, they get information via user actions as a mediator. Likewise, they cannot perform actions on environments, but they can offer advice to other parties.
2. Expert systems are not able to perform reactive and flexible actions.
3. The expert systems are not provided with social capability, i.e. collaboration and coordination.

In spite of the distinctions, these are expert systems that achieve real-time actions, they look very similar to agents. A salient example is the ARCHON infrastructure[2, 12].

7. AGENT ORGANIZATION

The organization indicates the method of communication among agents that are identified. Agents can be aggregated based on certain properties such as objectives, services, beliefs, and resources[1, 2]. Hybrids of these organizational ways are also potential. The widest approaches for arranging MAS are explained and depicted in **Figure 6**. As follows:

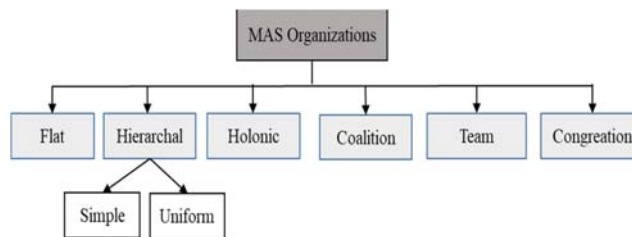


Figure 6. MAS organization.

7.1.Flat Organization

This is a fundamental of the structure where all agents are considered equivalent. In this organization, each agent may communicate instantaneously with other agents. There is no specific leader agent and each agent can contact any of the other neighboring agents. There are no proposed control of an agent on all others. This requires that the system must be closed in such a way that every agent must know everything about other agents in advance. In the case of an open system, an agent determining position methodology should offer a piece of the structure [2]. This organization has beneficial advantages because it provides agents the autonomy and openness of the agents-based system. It also permits changes in the dynamic arrangements of an organization and the environment. The openness may result in overheads in communication as well as the requirement of agent determining position mechanisms. The inference that an agent achieves in a flat organization and the computational overhead regarding other agents will increase significantly [9,10].

7.2. Hierarchical Organization

This organization is one of the basics and the oldest organizational structures. In this organization, the agents are organized in tree-like architecture. The agents at various layers of the tree have distinct autonomies. The data from lower layers can follow up higher to the agents in the higher levels[1]. The controlling signals may follow from higher levels to lower levels. **Figure 7.** depicts the hierarchical organization[2, 3].

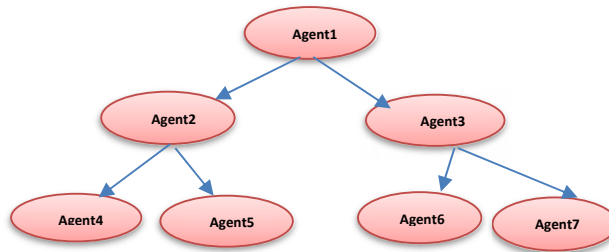


Figure 7. The Hierarchical Organization[2].

Hierarchical architecture can be divided into two types; simple and uniform structure based on the control distribution among agents.

1. Simple Hierarchy

In this hierarchy, the decision-making process and authorization are exclusive by the root agent of the hierarchy. The drawback of the simple structure is that the root agent can be single at a point of failure and may lead the system to fail[2, 9].

2. Uniform Hierarchy

In this hierarchy, the management and authorization are distributed between different agents to improve the reliability, and fault tolerance to have a smooth downgrading of the system if any failure happens. In the uniform method, multiple agents are authorized in the tree which means that in addition to the single-agent at a higher level of (root agent), all or certain parents agents can manage and control their child agents in the tree [2].

The decision-making process is taken by appropriate agents which have sufficient information. The decisions are transmitted upward in the tree in case if there is a contradiction among agents[2, 10].

7.3. Holonic Organization

In this organization, agents are arranged in several groups known as holons depending on certain properties, for example, the heterogeneity and the sensation capability [1]. In holonic system, an agent that is a singular component may have consisted of sub-agents surrounded by commitments. These sub agents are not surrounded by restrictions or by pre-determined rules, but just via commitments which indicate the agreed relation to all of the collaborative and shared agents in the holon[3, 9].

Each holon chooses a head agent that can connect with other agents or with the environment itself. The assignment of the head agent depends on the availability of resources, connection ability and internal structure of each agent. In the homogenous system, the choosing of a head agent is depended on the agent capability such that the formed holons may be aggregated further according to the beneficial advantages in forming the structures in order to form super

Holons[1, 2, 10]. **Figure 8.** Depicts the super Holon which is formed by aggregating two holons.

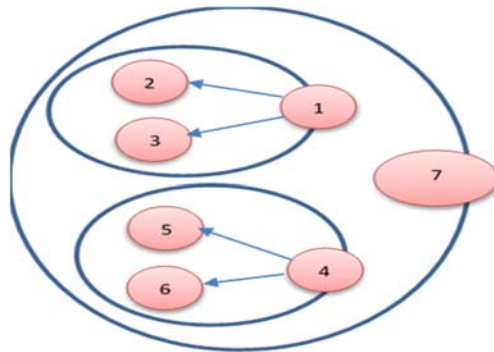


Figure 8. the structure of super holon and subholons [2].

Agents numbered 1 and 4 are the head agents of the two holons and are connected with agent 7 which is the super holon head. The structure is similar to the hierarchical architecture. However, the overlapping of agents that forms part of two distinct holons is permitted.

The advantage of a holonic organization is the agents' autonomy increases in the holon. The working of the holon offers a great deal of freedom when choosing the behavior[13]. The drawback of the holon is the lack of knowledge of the internal holon structure which makes it hard for other agents to anticipate the actions and behaviors of the holon [9].

7.4.Coalition

In this organization, the agents are arranged and congregated together according to their objectives into agent groups having similar objectives [1]. The agent groups are arranged for a short time to improve the performance of agents in the coalition. As a result, the agents may achieve their goals with lower overheads in comparison to other organizations where they don't have such grouping in which every agent can be part of multiple groups [3]. Furthermore, the agents destroy the group when they achieve their goals. In the coalition, the internal structure of agents is flat, although the structure such as hierarchical may be utilized.

This organization is appropriate when a group of agents with identical goals exists in the multi-agent system such that their cooperation joins them in achieving their aim[1]. In a flat structure within the coalition, it is probable to include a leading agent as a representative of the group. To enhance the shared knowledge within the group, the overlapping of agents inside the coalition is permitted [3, 9].

Due to the change in the behavior of the group, the coalition is hard to manage in environments that have dynamic behavior and it may require reorganizing agents to increase the performance of MAS. The coalition number has to be decreased to lower the related costs that result from creating and destroying a coalition depending on the threshold as the activity is measured and the group formalization can be predetermined [2, 9]. **Figure 9.** Illustrates the coalition.

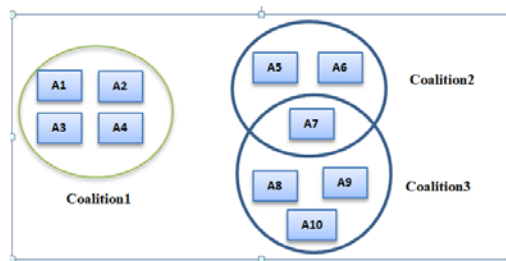


Figure 9. The coalition [9].

7.5. Team

The communication between agents in the team may be arbitrary and the objectives given to agents can differ depending on the enhancements which result from the team. The team objective can upgrade which results in changing the activities and authorizations of agents in a team[3]. To enhance its decision-making operation, every team can demand information from agents in other team structures. In achieving the team objective, a team can have an internal structure to enhance efficiency [10].

The greater team size is not advantageous under all circumstances. As a result, a trade-off must be done among the amount of information, number of agents and their learning abilities. Bigger teams provide an enhanced clarity of the environment and a great amount of associated information. However, encompassing the expertise of agents into a singular paradigm team is influenced. Agents' teams with smaller sizes provide vast learning abilities but lead to near-optimal performance because of the partial view of agents' environment. A compromise may be required in selecting the best team size which may maximize the cost much greater than coalition structure. **Figure 10.** Depicts a team depended on structure with a limited view.

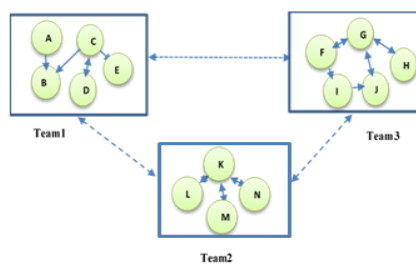


Figure 10. The system team structure with a limited view of the environment [3].

Teams 1 and 3 can communicate with each other and observe, but they cannot see teams 2, 4. The agents' actions and their activities are arbitrary within teams. In contrast to a coalition where agents are arranged together to achieve their jobs are arbitrary and differ within teams[3].

In contrast to the coalition, where agents are arranged to achieve their objectives, in the team agents they try to reach the team objectives. In this way, this structure is appropriate where several agents try to reach the same objective[3, 10].

7.6. Congregation

In this organization, agents in one position constitute a congregation to perform their activities. Agents in the congregation achieve actions that they cannot perform alone. Every

agent may join or depart the congregation and must be a member of one congregation at one time.

Congregations are constituted using agents with homogenous or complement properties. In a congregation, agents do not require to include a singular goal. The rate of agent achieving its goods in the congregation depends on other agents. This organization must have at least one agent. This structure is influenced such that every agent needs the other agents' resources to perform its actions[2, 3]. **Figure 11.** Illustrates congregation organization.

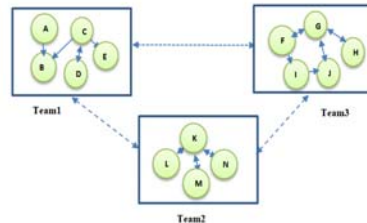


Figure 11. Congregation Organization [3].

8. Agent Architecture

Agent architecture is primarily made up of internal components that an agent and MAS can be built from, i.e. the data structure of agent, the actions that can be achieved on these data structures and the flow of control and communication among these structures[14]. A number of known agents' architectures can be illustrated in **Figure 12.** And explained as follows: -

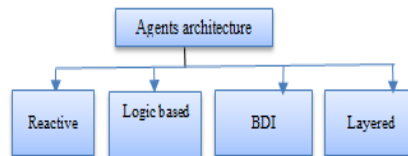


Figure 12. Agents architecture.

8.1.Reactive Architecture

Depending on stimulus responding methodology driven from sensory input, this architecture applies a decision-making mechanism for mapping of states to actions. They do not include the symbolic paradigm and do not use complex inference compared to logical architectures. The most common architecture of this kind is Brook's sub Sumption. The basic notion is that an intelligent action may be obtained without clear representations and inferencing which is offered by the symbolic AI method. The intelligence is merely an emergent characteristic of such sophisticated systems. The sub Sumption structure identifies levels of finite state machines and they are interconnected to sensors that transport information [9].

Figure 13. Illustrates the reactive architecture. It is observed that the agent perceives the environment and makes mapping of the environment state to action. Multiple actions can be suitable and the agent must select the most appropriate within the environment [9].

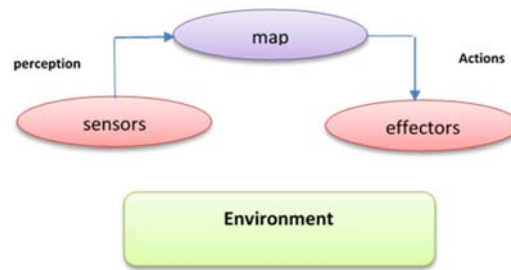


Figure 13. Reactive Architecture [12].

8.2.Logic-based Architectures

This symbolic structure stems from conventional knowledge-based methods where the environment can be represented and changed using symbols through inferencing techniques, and thus the agent behavior is based on the manipulation of the symbolic representation[9].

Belief, Desire and Intentions (BDI) Architecture

In this architecture, the beliefs expressed the assumptions and concepts of the agent world (The environment satiation in which it suited). Desires are the objectives that the agent wishes to perform. The agent has several desires. The agent can use both the beliefs and desires to select actions. To map and achieve the desires, this is what is meant by intention [16]. **Figure 14.** Shows the BDI architecture.

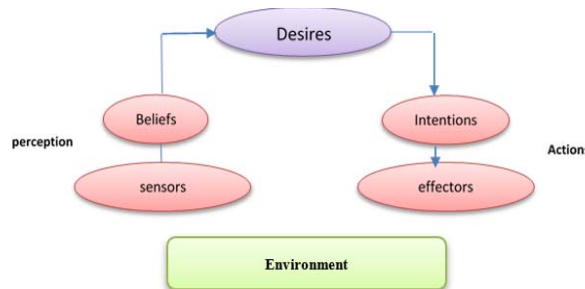


Figure 14.The BDI Architecture [16].

8.3.Layered Architecture

Layered (hybrid) architecture permits agents from both architectures: native and deliberative. To hold both types of action: Subsystems are organized as levels (or Layers) in the structure. In the Layered (hybrid) architecture, there are two kinds, horizontal and vertical layers[12, 15]. The layers are linked straightly to sensors and action results, as shown in **Figure 15.** Every layer acts like an agent[12, 15]. The vertical architecture may be divided into two control structures. In one pass structure, the control flows from the first layer that obtains sensory data to the last layer that reproduces actions result. In two-pass structures, the sensory input follows up the layer's series and flows down[12, 16].

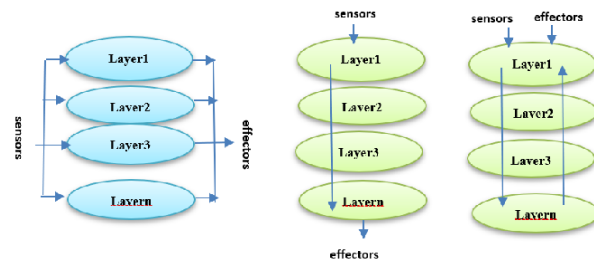


Figure 15. Layered Architecture.

9. AGENT TYPES

Per the characteristics they have, agents can be subdivided. Software agents are categorized depending on their tasks, for example, information collecting agent or agents that filter messages and emails. They can also be categorized depending on their control structure. Agents can be categorized on their sensitivity or affectivity of behaviors [4, 11].

Agents can be categorized based on:

- 1- Mobility, whether the agent is a static agent or dynamic one.
- 2- Existence of symbolic inferencing scheme, whether it is a deliberative reactive agent or not.
- 3- Presenting their primary features like collaboration, anatomy... etc.
- 4- Hybrid methods that incorporate two or more methods in a singular agent.
- 5- Secondary features like veracity, benevolence.... etc.

9.1. Collaborative Agent

These agents can behave reasonably and in an autonomous way in open and restricted environments. They have several features such as autonomy, responsiveness and social behaviors[17].

9.2. Interface Agent

They assist users while using computer applications via learning in cooperation with users and other agents. The main features are as follows: collaboration with user or other neighboring agents, and the ability of learning[15].

9.3. Mobile Agents

Mobile agents are such agents which use autonomous software and are capable of traveling in WANs and collaborate with other agents in performing tasks on the side of end-users. Their important features are mobility and cooperation to transmit information to other agents[17].

9.4. Information Agents

They can manipulate the large quantity of available information in several sources which are scattered. The main characteristics of these agents are mobility and they can be static dynamic agents. They can be non-collaborative or have the social ability with other agents[11].

9.5. Smart Agent

They have similar main features such as learning and collaboration with other agents[4].

9.6. Hybrid Agent

In order to increase the strong points and decrease the drawbacks of the most associated methods, these agents incorporate multiple agents techniques in one singular agent[17].

9.7. Reactive Agent

Reactive agents are constrained in their activities and they cannot regulate among themselves and realize certain goals. When events occur in their environment, they merely react to them. However, they can play a function in introducing intelligent behavior[4]. The agents are not necessarily to be independently intelligent, they can run with other agents cooperatively to solve sophisticated tasks[12].

9.8. Cognitive Agent

Cognitive agents can introduce intelligent behavior in some ways. In this approach, intelligent agents have aims and can find solutions and perform them. They can regulate their actions and utilize complex communication approaches[3]. Using their knowledge, the agents can interact with their environment. They can also upgrade and modify their states using learning techniques to instruct and lead their actions and make decisions. They have more flexibility comparing to reactive agents[12].

10. Communication in MAS

Three common methods of agent communications are explained below:

10.1. Speech Act

The theoretical idea of speech act deals with agent communication like actions or behaviors. This theory begins with the research of Austin. He remarked that a special kind of utterance (known as speech act) had the features of behaviors or actions, such that they can change the physical world situation in a method similar to physical behavior. For example, if someone raises a cube from a table, then the physical environment has altered somehow. If someone says "I announce you like a man and a wife", then this statement influences the world by determining new identifications and changes[5].

Austin declared several performatives, which means various kinds of speech acts. Examples of such speech acts are, inform, ask, stream, etc. Austin also indicates the conditions and requirements needed for completing or performing the performatives. He pointed out them as feasible conditions. Moreover, an agent can behave like a speaker that introduces utterance to alter the ideas and assumptions of a listener (or hearer). Agents achieve utterance by performing actions that are observed as an utterance in accordance to grammar[2, 5].

10.2. Message passing

It is also called local communication. In this approach, agents send messages to other agents. The information movement is in two directions which makes a distributed structure and minimizes the possibility of failure of central agents [2]. **Figure 16.** Shows the message passing method. To communicate with neighbor agents, the agents utilize peer-to-peer or broadcast communication. Agent 1 can immediately communicate with Agent 2 by knowing the address of Agent 2. This approach is used in peer-to-peer communication. In a broadcast approach, Agent 1 communicates with all other agents by sending them a message. The agents should use a consistent structure to assure that the message is interpretable[17, 18].

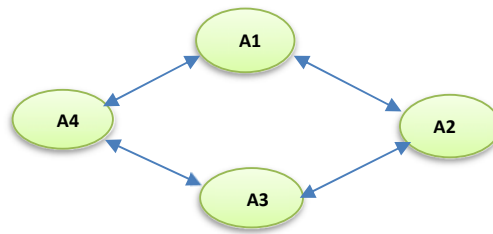


Figure 16. Message Passing [2].

In this approach, agents may cooperatively share a repository that contains data with other agents. The repository offers effective storage and retrieving of data for agents. **Figure 17.** illustrates the blackboard communication[2, 3]. Every agent can save its data and information on the blackboard which can be read by neighbor agents. The blackboard utilizes a controlled knowledge to manage agents accessing them. Every agent may reach some data in the control knowledge[2, 5].

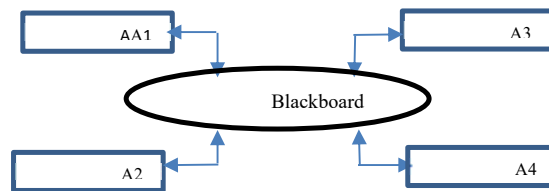


Figure 17. The Blackboard Architecture.

11. Agent Communication Language ACL

Because of the increase in agents' number and their heterogeneous groups, a shared structure must exist to assist the appropriate cooperation and data sharing among agents. The semantics of messages is crucial to assure that communication among agents has the same meaning of exchanging information. This shared structure is offered by agent communication language[3, 5]. The designing concepts of ACL should include:

- Shared interaction language (which is also called a protocol) that can be realized by other agents.
- A common ontology such that the sent messages include the same meaning and follow of agents' semantics.

ACL must ensure coherence which means that the same performatives must contain the same semantic. ACL should also ensure affinity which means that all messages use similar words to say similar things[16]. Furthermore, ACL is needed to facilitate agent communication with other agents in order to communicate their aims, desires, and beliefs. ACL permits agents to enhance other agents to perform goals, to observe their running, to report their progress, to refuse to perform tasks and to agree for achieving goals for neighboring agents[18].

ACL is modeled depending on the statements sharing determined hypothesis definitions etc. These declarative sentences should be meaningful to incorporate a variation of the pieces of information[2]. The declarative sentences can be short in length and concise because the long statements may influence the communication cost and the damage of information. The declarative sentences are required to be simple to permit using higher language. To match all these needs in ACL, the DARPA knowledge-sharing effort (KSE) was constituted in 1990

with the motivation of finding and developing protocols for knowledge exchange between autonomous agents[2].

The two main important achievements of KSE are:

- Knowledge Query and manipulation language (KQML) which is an external language for message communication. It represents an envelope which covers the message. KQML is not interested in constructing the content of messages.
- Knowledge interchanges format (KIF). It is intended to represent the information and knowledge concerning the certain domains. It was designed to construct the content of the message[5, 18].

In general, the ACL contains three parts: a vocabulary, internal language, and external language. The internal language is used to map the communication data into logical pieces of information that are realized by other agents. KIF is one of the internal languages used in messages[5]. The representation used in the inner (internal) language must be abbreviated, precise and unambiguous as the recipient should stem from the original meaning. ACL manages a massive repository of vocabulary and makes it open-ended in order to modify it with new words to increase its functionality. Several ontologies must be included in the repository to use it in different domains[18].

12.1. Knowledge Interchanges Format (KIF)

Is a popular internal language that can be used to represent the features of a certain domain? KIF was invented to represent message content and it depends primarily on first-order predicate logic in Lisp format. KIF can encode simple information, restrictions, rules, and Meta level data that can help in making decisions. KIF cannot be used only to exchange information because there are implicit pieces of information that should be embedded in the message. The receiver can interpret the KIF message with little information about the sender[5, 16].

The agents can be represented by using the following KIF:

- The features of concepts in the domain
- Relations among concepts in the domain.
- General characteristics of the domain.

KIF proposes a fixed logical structure that includes the basic connectors that are used in first-order logic like (and, or, not) and the quantifiers used for universality as (for all, exists). KIF also offers a vocabulary used to represent concepts like numbers, and letters. Relationships are also contained in KIF to represent these concepts like "less than" relation. Using this structure, KIF can define concepts, and their relations [5, 16, 18].

12.2. Knowledge Query and Manipulation Language (KQML)

KQML is a language for messages used in agent communication. It is an external language that determines a shared form for agent messages. KQML is also considered as message protocol to aid easy communication. Every message can contain performatives and several parameters (feature values)[5].

KQML is considered as a high-level of language for communicating information as messages among agents. It has nearly three dozen speech acts or performatives that agents can utilize in order to constitute sophisticated cooperation and negotiation methodologies[16]. Typical examples of KQML and KIF languages are explained as follows.

Ask: content (cost service-1 ?cost)

- : language prolog
- : ontology services-domain
- : in reply to quote service.
- : receiver customer-3
- : sender customer services

KQML can contain three layers: The first layer is communication layer which includes the sender and receiver information and an identifier for a query, and a message layer which determines the task to be achieved (in the above example, the sender agent asks the cost of service from the receiver agent). Finally, the message contains a content layer that gives all the necessary information to achieve the query[18, 19]. **Figure 18.** Depicts three layers of KQML message.

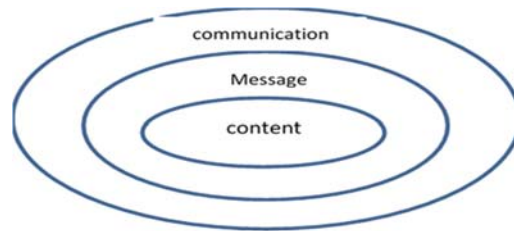


Figure 18. The three Layers of KQML Message [18].

The communication layer is the lowest layer that deals with packets, and the layer uses the protocols TCP/IP, UDP for packet transmission. The message layer contains the task (performative) that is intended to be performed. The content layer determines the language used, the ontology of domain and other useful information in performing the performative message[18].

KQML has several drawbacks. Firstly, the KQML specification is ambiguous and dubious as it lacks a fixed, accurate semantics. Secondly, KQML undergoes from lacking some performatives and some of them are miss-determined. KQML misses some kinds of performatives, commissives, which allow an agent to perform a commitment or consensus with other agents. Commitments are rather important and multi-agent systems can use them to cooperate with other agents. Finally, the list of performatives of KQML is very huge. These shortcomings lead to developing a new language standard that tackles these problems[5, 18].

The FIPA ACL

FIPA, acronym for (Foundation for Intelligent Physical Agents) is a developing ACL standard for agent communication. It is similar to KQML in representing an external language for messages. FIPA ACL includes nearly 20 performatives and does not consider any content language for messages[5].

FIPA ACL structure and message features are very similar to KQML. One of the differences between the two languages is the performatives they offer. In addition, FIPA ACL enjoys global semantics which is offered. According to SL formal language, SL allows expressing beliefs, intentions, and behaviors that an agent can perform[16, 18].

The FIPA ACL semantics conform each message to SL formula defining a restriction (condition) that the sender must achieve (or fulfill) in order to conform to FIPA ACL. This restriction is named as feasibility condition[19].

12. Ontology

In addition to ACL, shared and known concepts between agents are fundamental for agent communication. Ontology is needed to express the information and knowledge from different

domains. Without using a familiar ontology which includes the words, terms, and relations relating between them, the ACL still only develops a syntactic language for messages [6, 12].

There are many languages that can be used to define and express ontology such as KIF. By KIF, we can define the features of concepts in the domain and relations between them. Software programs can be used to permit a user to create ontologies using KIF languages. An example of this is the ontology server which is a service on the web that supports a shared paradigm to store the shared ontologies refined by various groups of developers.

KIF is much dependent on first-order logic that supports it with a precise meaning and makes it demonstrative and meaningful. There are several other languages that can be used to express ontologies such as XML (Xtensible markup language) [6].

13. Java Agent Development Environment (jade)

JADE, an abbreviation of (JAVA Agent development environment), is a software platform that simplifies the creation and development of agent-based systems for homogenous entities as a method for assisting researches and establishing applications. It conforms to the foundation for intelligent physical agents (FIPA) standard. JADE uses a group of tools and APIs that simplify the troubleshooting and distributing phase[6, 12].

JADE is considered as middleware developed for establishing multi-agent systems using the architecture of peer to peer. The data, recourses, controls are dispersed on machines in the network. The agent framework may be scattered among computers (which are not necessary to contain similar operating systems) and the arrangement and settings may be altered and managed by a GUI [20, 21].

JADE framework enjoys a higher degree of functioning in the multi-agent system programmed in Java. The JADE objective is to facilitate the development of agent-based system as well as providing a global group of services for developing agents. JADE provides some properties to the agent developers[21].

1. The agent platform conformed to FIPA standard, contains the standard types of agents triggered when the framework is started. These agents are (AMS, abbrev. Of Agent Management System, DF, abbrev. Of Directory Facilitator and ACC, abbrev. of Agent Communication Channel)
2. Dispersed agent framework. The agent framework may be divided into machines. For each machine, there is only a need for a single Java application and single JVM.
3. JADE supports API to send and receive messages among agents.
4. JADE provides a library of FIPA protocols used for interaction among agents.
5. JADE allows the mobility of agents in the framework.
6. GUI is to control and maintain the agent framework. The admins of the framework can create new agent actions, delete and manipulate.

14. Applications of MAS

There are several applications of MAS in different domains, we list below some of them:

Web Advertisement

The researcher in [22]. Proposes a methodology for customizing and personalizing web advertisements using the mining of contents and web. The objective of the proposed personalized system is to offer advertisements that are appropriate for user interests. To perform this objective, more data are manipulated inconsistent vector space: the contents of

advertisers and the publisher's site, the previous and current behavior of the user. Agent-based system is working on the publisher's website to arrange customized advertisements. Each collaborating agent has a specific job. The proposed system can suggest hyperlinks in adaptive methods. The system is scalable and can accommodate new functionalities as complicated scheduling methods and personalization using knowledge which are stemmed from buying history in addition to the ratings of products. The proposed system introduces a variety of updating processes. Another research about web services[23]. Is compared and discussed in **Table 1**.

15. Cloud Computing

Cloud computing is used to allow access to massive amounts of computation resources online. The clients can use the information and retrieve them and utilize services using computers and smart devices at any time to perform computation processes and to reach their data. The developers have realized the need for a system that utilizes the cloud with enhancing its performance. MAS offers an ideal method for complex systems that are altered dynamically. MAS can resolve problems using a decentralized approach such that multiple agents collaborate to achieve a task.

In the work of AlWadan [24]. A new infrastructure is presented which includes MAS that uses cloud computing to help in deciding the resources and utilizing the overall ability of the cloud. This architecture can also monitor the user's tasks which are processed in the cloud. The JADE framework is used to implement the proposed techniques. The simulation is performed with two experiments with a different number of jobs that are transmitted to the cloud computing system. From the performed experiments, the proposed techniques can achieve the following:

- Introduce a novel system that comprises MAS with a cloud paradigm to assist in selecting the appropriate resources and help in performing negotiation among cloud suppliers and users to utilize the overall cloud power.
- The proposed technique can watch and supervise user tasks during their processing.

Cloud computing can offer many advantages to universities and institutes. It provides resources and services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Today, colleges and universities have used the cloud and its potential power for innovation. Depending on the benefits of cloud computing and the MAS method to assist e-learning, Mohammed et al in [25]. Proposed a hybrid method that comprises MAS and cloud computing to formalize a new layer named Smart-CLS, where CLS is an abbreviation of Cloud Learning System. The proposed approach offers three items:

- Cloud computing provides electronic learning as a service (LaaS)
- MAS technique must be included as a service in the cloud
- A new level called Smart-CLS is added to offer services in any time and in all places for all the participants in the learning method.

The proposed system is a software layer that represents a platform for interactions and communications with the system. The technique has the following advantages:

- Support GUI agent
- Simplify the MAS distribution in the cloud computing

The results of the system experiments have authorized and validated the use of Smart-CLS and have shown that the application has greater importance in universities and colleges

with the goal of offering assistance in modern methods for learning with students' satisfaction of using it.

16. E-Commerce

Agents provide the facility to alter e-commerce by assisting traders to buy products depending on their preferences. E-commerce systems grant values to clients by offering custom experiences. In [26]. The researcher aims to develop a cooperative MAS using an e-commerce paradigm that supports autonomy and personalization. The system consists of three types of agents: mall, seller and buyer. The basic concept of the technique is to obtain a broker which identifies the buyer's requirements using default parameters that are used to solve the case of finding items of interest. The proposed method targets to provide responses to requests and to offer items according to the user's needs.

The proposed technique is a peer to peer system. The peers have the capability to connect in a reliable way and share data. The proposed method permits peers to be directly connected without using a center server which makes them reliable and more secure. There are three types of agents: Mall (MA), Buyer (BA) and Seller (SA).

The Seller agent has data that is needed to be offered to the buyer agent. The buyer agent needs some information about the request of its client. The Mall agent presents a list of sellers that meet the clients' needs by comparing the need with the properties of the product that exists in the shared database in the Mall agent. Recent e-commerce techniques used a centralized approach as a server which they represented at a point of failure. If the server fails, the whole system will fail too. Although when the Mall agents fail, the Buyer agent can negotiate with the seller agents. The proposed framework avoids the shortcomings of e-commerce paradigms by owning a peer to peer communication, which is an efficient medium of communication. The results prove that the proposed system permits locating the items in a way coherent with customers' requirements. The results show proves the hypothesis, that the cost of the result is improved and also confirmed the reality that the system acts as a peer to peer model.

The authors in [27]. Provide the basis for sentimental negotiation and the job of emotion in making decisions. In this approach, the authors presented a MAS with an e-commerce paradigm. To grasp the effect of negotiation, it is necessary to construct a model that comprises human emotions. Such a model has the ability to merge with the process of negotiation like humans. The agents included in the dual negotiation for goods must include some representation of users' interests. The agents are able to grasp and clarify knowledge like this. The proposed techniques have three main elements: the client, the provider, and the community. It is such a novel model in which emotions are incorporated into each issue. From consumer knowledge obtained, negotiation to consumer evaluation, and provider marketing mechanisms. The proposed model is capable of grasping the reasonable and emotional manners, it is also capable to learn the interests of consumers. It can collect and utilize the negotiation operated in the social opinions, which are emotional, and provide a view of the emotional situation to the participant of negotiation exhibiting internals.

17. Image Segmentation

The proposed technique in [28]. Provides a MAS methodology for segmenting images. This technique divides the image space into a group of sub-spaces as parts of the image such

that, numerous agents can determine the object boundaries in order to follow them. The detection of the following agents can adopt a method of detection and keeps the track of boundaries depending on the regional features that they develop. The aggregated information is sent to the supervising agents which process them with gathering the information released by detecting the following agents in order to modify tables including the segmenting parameters and to improve global management methods of detecting the following agents (such as creation, destroying of detecting-following agents). The supervising agent exists at the higher layer of the system hierarchy. The proposed system is implemented using the MadKit framework which improves the performance and makes the results more precise because of the simultaneous cooperation running of tasks.

The proposed system was experimented using a database containing three thousand images which had a lot of distinctions and details. The proposed system produces acceptable and satisfying results in accuracy and in running time. The adequate results illustrate the capability of the agents to segment different kinds of images. The proposed method enjoys many benefits regarding other approaches due to selecting the detection method by detecting the following agents' properties of their regions in such a way that each agent selects the determining method in accordance with the properties of its zone.

Today, image segmentation approaches are important processes in different fields. Image segmentation can be considered as an arguable problem because of many aspects. First of all, image segmentation methods are dependent on the problem. Secondly, segmentation techniques on medical images have constraints and limitations because similar images enjoy similar nature and patterns between objects. In [29]. A paradigm for extracting concurrently numerous objects from CT images uses previous or known knowledge. The local agent labels the sub-image. The Moderator agent examines the results of agents' work to generate the final image that is segmented. The experiments of CT images show segmentation precision results of 91% and the effectiveness of 7 seconds. The proposed technique is based on MAS such that each agent is identified with some characteristics to achieve the segmentation process and segment images are very complex in nature. The agent tries to discover a suitable mark for every pixel in the image. The Moderator agent creates several images inside the image. The image is partitioned to many sub-images, in such a way that every agent works on a sub-image and labels every pixel using given knowledge. The local agent labels every pixel in the sub-image. In the last, the moderator examines the result of agents to provide the last segmented image.

The proposed system can work automatically; it can work without intervention from the user in the segmentation process. The benefit of the proposed approach is to segment the image into several zones in parallel. This means that there are several zones such that the properties of each zone are different. This technique is fast regarding other methods with the following results:

1. The system provides good results in the precision of segmentation; the average precision is 91% for every zone in the image.
2. The system achieves satisfying outcomes in execution time; the average execution time is about 8 seconds.
3. The ability of the system to perform segmentation on an image into a distinctive zone concurrently.

18. Computer Networks

Gustavo Isaza et al propose in [30]. A MAS for managing the network. The proposed model defines agents' interaction to perform the goals and needs of the MAS architecture. The agents have the characteristic of adaptation, obtaining knowledge in order to perform decision making in accordance with the actual network situation that is expressed in the SNMP equipment. The optimal situation of the network is identified by the user entered, which includes functionality values and other parameters. The architecture of agents maximizes the cooperation, autonomy and powerful work in heterogeneous frameworks in-network supervising. The proposed technique introduces MAS for managing network, which utilizes four agents that achieve certain activities and communicate with each other through messages. The messages have conceptualization with certain domain ontology. The agents get information from the devices and equipment of the network by using SNMP, check it and identify possible anomalies that are discovered. When the agents found such anomalies, the proposed system alert uses warnings or alarms to the managers in order to respond appropriately.

The MAS allows forming and constructing the environment for network management to be autonomous, adaptive and transmitting conditions of the environment where the agents run with minimum intervention of the user. In accordance with the policies restrictions and thresholds, the running of the proposed system gives expected results. The impact of MAS execution in order to monitor and manage the network can be detected if the position policy is crushed. The agents' detection and behavior of using message passing permit to perform a scan to the network; later, the manager agent produces an alert that is saved and transmitted to the interface agent. Finally, the interface agent is responsible for presenting it to end-users. The proposed technique raises the performance of supervising the network and decreases the policies. It permits presenting learning methods that execute the paradigm in an autonomous network managing mechanism. The use of MAS ontology in SNMP can be considered as information representation dealing with problems in the dispersed network managing systems. MAS platform is an ideal solution for dispersed network management. This technique can be contradicted with the conventional management methods according to the responding time and speed by using the performed paradigm. MAS is suggested as a solution to the network management problem. The proposed technique expands the MAS structure to be integrated with the SNMP agent.

Table 1. Explains the contributions, architectures, application and limitations of each one of the above-mentioned MAS applications.

Ref.	Applications	Architectures	Contributions	limitations
Kazienko, [22]	Web Advertising	Software and web server	MAS based on web advertising: scheduling and personalization	More extension to recognize the behavior of users
Maximilien and Singh, [23]	Web Services	Network and software	The MAS framework of QOS and trust is based on ontology.	limited study regarding the behavior of the agent
AlWadan, [24]	Cloud computing	JADE and distributed	Identification of best resources in the creation of a negotiation method between cloud providers and users	Simple design for a user process

Ouidad and Kamoun [25]	Cloud computing	JADE and a set of classes and graphical tools	A new layer called Smart Cloud Learning System (Smart-CLS) must be set, in order to provide services anywhere at any time and using any devices, for all actors in a learning session	Lack of implementing Smart-CLS in Massive Open Online Courses (MOOCs).
Helmy [26]	E-commerce	Distributed	MAS e-commerce framework that enables autonomy, proactivity, and customization, including a smart mall agent, seller agent and buyer agent with their own profiles.	Limited scalability
Jascanu et al. [27]	E-commerce	Logical and reactive	Place of human counterpart "emotional system"	No test for Performance
Mohammed and Said, [28]	Image segmentation	Distributed based on groups of agent	Adaptation of the agents of segmentation in front of a big variety of images	Limitation in recognition approach
Chitsaz and Seng, [29]	Medical Image	Network-based on local and moderator agent	Image segmentation is based on multi-agent system	Limit constraints of the noise image
Isaza et al. [30]	Network Management	Simple Network Management Protocol	Comparing the GAIA, INGENIOUS and MASE agent and Integration of Ontology for the MAS in SNMP	Ignore the fundamental agent properties in designing such as intelligent and autonomous
Scalar et al. [31]	Cyber-Physical Production System	Logical Architecture and Software Architecture	The design and implementation for an agent-based CPPS network	Does not consider security issues in design and flexible autonomous facilitates

19. Conclusion

In this survey, we presented a comprehensive study concerning different issues of Multi-agent systems which helps new researchers and practitioners to acquire the knowledge for basic concepts and as insightful resources in MAS. As we have shown throughout the study, MAS is a unique software architecture, distinguishable from other architectures, which provides solutions to a specific family of computational problems. Firstly, we offered definitions and features of agents and MAS. We explained their environments, organizations, and architectures. We have demonstrated communication strategies and ACLs that are used by the agents to connect with other agents. Eventually, we have discussed, compared and summarized the role of MAS in some of the most important applications such as web advertisements, cloud computing, e-commerce, image segmentation and computer networks, and how can these applications deal with MAS subject and their findings with their contributions and limitations in each application.

References

1. Balaji, P.; Srinivasan, D. An introduction to multi-agent systems, in Innovations in multi-agent systems and applications-1. *Springer*.2010, 1-27.
2. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-agent systems: A survey. *IEEE*.2018, 6, 28573-28593.

3. Glavic, M. *Agents and multi-agent systems: a short introduction for power engineers*, **2006**.
4. Georgakarakou, C.E.; Economides, A.A. Software agent technology: an overview application to virtual enterprises. *Agent and Web Service Technologies in Virtual Enterprises*, **2003**.
5. Wooldridge, M. *An introduction to multiagent systems*.**2009**, John Wiley & Sons.
6. Al-Karkhi, A. *Task Recovery in Self-Organised Multi-Agent Systems for Distributed Domains*.**2018**, University of Essex.
7. Mostafa, S.A.; Ahmad, M.S.; Mustapha, A.; Mohammed, M.A. A concise overview of software agent research, modeling, and development. *Software Engineering*.**2017**, 5, 1, 8-25.
8. Abbas, H.A.; Shaheen, S.I.; Amin, M.H. *Organization of multi-agent systems: an overview*. arXiv preprint arXiv:1506.09032, **2015**.
9. Chin, K.O.; Gan, K.S.; Alfred, R.; Anthony, P.; Lukose, D. Agent Architecture: An Overviews. *Transactions on science and technology*.**2014**, 1, 1,18-35.
10. González-Briones, A.; Fernando, D.; Mohamad, M.S.; Omatu, S.; Corchado, J.M. Multi-agent systems applications in energy optimization problems: A state-of-the-art review. *Energies*.**2018**, 11, 8, 1928.
11. Iqbal, S.; Altaf, W.; Aslam, M.; Mahmood, W.; Ghani Khan, M.U. Application of intelligent agents in health-care. *Artificial Intelligence Review*.**2016**, 46, 1, 83-112.
12. Anaekwe, J.C. *Intelligent Multi-Agent Online Examination System*. February, **2015**.
13. Abar, S.; et al. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*.**2017**, 24, 13-33.
14. Collier, R.W.; O'Neill, E. MAMS Multi-Agent MicroServices. in *Companion Proceedings of The 2019 World Wide Web Conference*, **2019**.
15. Shehory, O.M. *Architectural properties of multi-agent systems*.**1998**, Carnegie Mellon University, The Robotics Institute.
16. Soon, G.K.; et al. A review on agent communication language, in *Computational Science and Technology*.**2019**, 481-491.
17. Mian, N.A.; Ahmad, F. Agent based Architecture for Modeling and Analysis of Self Adaptive Systems using Formal Methods. *International Journal of Advanced Computer Science and Applications*.**2018**, 9, 1, 563-567.
18. Kaur, A.; Jain, S. Kqml-from scenario to technology. *International Journal of Advanced Studies in Computers, Science and Engineering*.**2018**, 7, 3, 30-34.
19. Juneja, D.; Singh, A.; Singh, R.; Mukherjee, S. A thorough insight into theoretical and practical developments in multiagent systems. *International Journal of Ambient Computing and Intelligence (IJACI)*.**2017**, 8, 1, 23-49.
20. Priyadarshana, H., Hemapala K.T.M.U, Wijayapala W.D.A.S. Developing Multi-Agent Based Micro-Grid Management System in JADE. in *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC) IEEE*, **2019**.
21. NASIR, N.; ITO, T.; HASHIM, A.Y.B. *Agent-Based Communication in JADE Platform for Manufacturing Processes*.**2015**.
22. Kazienko, P. Multi-agent system for web advertising. in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*,**2005**.

23. Maximilien, E.M.; Singh, M.P. Multiagent system for dynamic web services selection. in *Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS)*, **2005**.
24. Alwadan, T. Cloud computing and multi-agent system: Monitoring and services. *Journal of Theoretical and Applied Information Technology*, **2018**, 96, 05.
25. Bousmah, M.; Labouidya, O.; El Kamoun, N. Design of a cloud learning system based on multi-agents approach. *International Journal of Advanced Computer Science and Applications*. **2015**, 6, 3, 20-26.
26. Helmy, T. Collaborative Multi-Agent-based e-Commerce Framework. *Int. J. Comput. Syst. Signal*. **2007**, 8, 1, 2-12.
27. Jascanu, N.; Jascanu, V.; Nicolau, F. *A new approach to E-commerce multi-agent systems*, **2007**.
28. Mohammed, R.; Said, A. An adaptative multi-agent system approach for image segmentation. *International Journal of Computer Applications*. **2012**, 51, 12.
29. Chitsaz, M.; Woo, C.-S. Medical image segmentation using a multi-agent system approach. *Int. Arab J. Inf. Technol*. **2013**, 10, 3, 222-229.
30. ISAZA, G.; et al. Network management using multi-agents system. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*. **2012**, 1, 3, 49-54.
31. Salazar, L.A.C.; et al., Platform independent multi-agent system for robust networks of production systems. *IFAC-PapersOnLine*. **2018**, 51, 11, 1261-1268.