CHAPTER 2

REVIEW OF LITERATURE

According to “Intelligent agents:” [1] describes that the concept of an agent has become important in both artificial intelligence and mainstream computer science. The aim of the paper is to point at what we perceive to be the most important theoretical and practical issues associated with design and construction of intelligent agents. Agent theory is concerned with the question what agent is; and use of mathematical formalisms for representing and reasoning about the properties of agent. Agent architecture can be thought of as software engineering models of agents; researchers in this area are primary concerned with problem of designing software or hardware systems that will satisfy the properties specified by agent theorists.

According to “Software agents” [2] it describe the software agents as fastest growing area of Information Technology. They are being used, and touted, for applications as diverse as personalized information management, electronic commerce, interface design, computer games, and management of complex commercial and industrial processes. It describes properties of agents & fields where agents can be used.

According to “Software Agents as Next Generation Software Component” [3] it describes as new classes of applications demand flexible, intelligent solutions for increasing volume of Business-to-Consumer (B2C) and Business-to-Business (B2B) Internet traffic problems. Distributed software agents offer great promise, building on increasingly pervasive message-based middleware and component technology, Java, the Extensible Markup Language (XML), and the Hypertext Transfer Protocol (HTTP). Agents are specialized kinds of components, offering greater flexibility
than traditional components. Agents rely on an infrastructure that provides services and mechanisms, allowing agents to have simpler interfaces and be more composable. It also defines basic structure of agent.

According to “Designing Reusable Behaviors for Information Agents” [4] it describes specification and effective implementation of various classes of computational agent behaviors. One such class of behaviors involves collecting and supplying information to other computational agents or humans. We call an agent that exhibits such behaviors an information agent. Information agents can play an important role in many larger mixed human- and computational- agent organizations. The dominant domain level behaviors of an information agent are: retrieving information from external information sources in response to one shot queries requests for periodic information monitoring external information sources for the occurrence of given information patterns, called change monitoring requests. Information originates from external sources. Because an information agent does not have control over these external information sources, it must extract, possibly integrate, and store relevant pieces of information in a database local to the agent.

According to “Task-based Multi-agent Coordination for Information Gathering” [5] describes of developing distributed collections of intelligent information agents that cooperate asynchronously to perform goal-directed information retrieval and information integration in support of various tasks, such as finding information about people on the Internet, managing calendars and making arrangements to host visitors in an academic environment. The task of hosting a visitor involves arranging the visitor's schedule with faculty that match the interests that the visitor has expressed in his/her visit request.

According to “Middle-Agents for the Internet” [6] describe that Like middle-men in physical commerce, middle agents support the flow of information in electronic commerce, assisting in locating and connecting the ultimate information provider with the ultimate information requester. Many different types of middle-agents will be useful in realistic, large, distributed, open multi-agent problem solving systems. These include matchmakers or yellow page agents that process advertisements, blackboard agents that
collect requests, and *brokers* that process both. The behaviors of each type of middle-agent have certain performance characteristics—privacy, robustness, and adaptiveness qualities—that are related to characteristics of the external environment and of the agents themselves. For example, while brokered systems are more vulnerable to certain failures, they are also able to cope more quickly with a rapidly fluctuating agent workforce and meet certain privacy considerations.

According to *“Methods for Task Allocation Via Agent Coalition Formation”* [7] it describes that task execution in multi-agent environment may require cooperation among agents. Given a set of agents and a set of tasks which they have to satisfy, we consider situation where each task should be attached to a group of agents that will perform the task. Task allocation to groups of agents is necessary when tasks cannot be performed by a single agent. However it may also be beneficial when groups perform more efficiently with respect to single agent. It represents the algorithms with low bounds and with low computational complexity.

According to *“A Scalable Agent Location Mechanism”* [8] describe that large scale open multi-agent systems where agents need service of other agents but may not know their contact information require agent location mechanisms. Solutions to this problem are usually based on middle-ware such as matchmakers, brokers, yellow-pages agents and other middle agents. The disadvantages of these are that they impose infrastructure, protocol and communication overheads, and they do not easily scale up. It suggests a new approach to agent location which does not require middle agents and protocols for using them.

According to *“A Planning Component for RETSINA Agents”* [9] describe that In the RETSINA multi-agent system, each agent is provided with an internal planning component—the RETSINA planner. Each agent, using its internal planner, formulates detailed plans and executes them to achieve local and global goals. Knowledge of the domain is distributed among the agents, therefore each agent has only partial knowledge of the state of the world. Furthermore, the domain changes dynamically, therefore the knowledge available might become obsolete. To deal with these issues, each agent’s
planner allows it to interleave planning and execution of information gathering actions, to overcome its partial knowledge of the domain and acquire information needed to complete and execute its plans. Information necessary for an agent’s local plan can be acquired through cooperation by the local planner firing queries to other agents and monitoring for their results. In addition, the local planner deals with the dynamism of the domain by monitoring it to detect changes that can affect plan construction and execution. Teams of agents, each of which incorporates a local RETSINA planner have been implemented. These agents cooperate to solve problems in different domains that range from portfolio management to command and control decision support systems.

According to “Agent-Based Support for Human/Agent Teams” [10] describe an interface agent, MokSAF, which facilitates time-critical team-planning tasks for teams of both humans and heterogeneous software agents. This agent assists in the formation of teams of humans (via other MokSAF agents) and task agents that can autonomously perform team subtasks. It provides a suitable interaction mechanism to instruct the various task agents in the team; and, by monitoring the human’s progress, reallocate or modify the sub-tasks if the human fails to achieve that subtask. A military domain has been used to investigate this interface agent. The task consists of three military (human) commanders that each assemble a platoon, and plan routes so that all three platoons arrive at a given rendezvous by a specified time. An experimental study has been conducted to evaluate MokSAF and the assistance provided by one of three different task agents, and the results summarized.

According to “Avoiding Resource Conflicts in Intelligent Agents” [11] describes that intelligent agent should be rational, in particular it should at least avoid pursuing goals which are definitely conflicting. In this paper we focus on resource conflict in agents that use a plan library organized around goals. It characterizes different types of resources and define resource requirements summaries. We give algorithms for deriving resource requirements, using resource requirements to detect conflict, and maintaining dynamic updates of resource requirements. It also discusses ways of resolving resource conflict. Our approach does not represent time; rather it keeps resource summaries
current. This enables an agent’s decisions to be made on the basis of up-to-date information and allows us to develop efficient runtime (online) algorithms.

According to “Task Characteristics and Intelligent Aiding” [12] describes the interactions between task characteristics and human agent interfaces in a team rendezvous route-planning task. The agents include an interface agent and two different task agents that perform similar tasks. The MokSAF interface agent links an Artificial Intelligence (AI) route planning agent to a Geographic Information System (GIS). Through this agent, the user specifies a start and an end point, and describes the composition and characteristics of a military platoon. Two aided conditions and one non-aided condition were examined. In the first aided condition, a route-planning agent (known as the Autonomous RPA) determines a minimum cost path between the specified end points. The user is allowed to define additional "intangible" constraints that describe situational or social information that should be considered when determining the route. In the second aided condition, a different agent, the Cooperative RPA, uses the same knowledge of the terrain and cost functions available to the Autonomous RPA, but restricts its search to paths within regions drawn by the user. In the unaided condition, Naive RPA, the user draws the route manually, then submits it to be tested against the terrain and cost functions for feasibility. Both aided conditions are superior to the control but differ in their relative effectiveness by scenario. This paper examines the varieties of challenges faced by commanders in two scenarios and relates them to the differential effectiveness of the agents.

According to “Agent-Based Team Aiding in a Time Critical Task” [13] evaluate the effectiveness of agent-based aiding in support of a time-critical team-planning task for teams of both humans and heterogeneous software agents. The team task consists of human subjects playing the role of military commanders and cooperatively planning to move their respective units to a common rendezvous point, given time and resource constraints. The objective of the experiment was to compare the effectiveness of agent-based aiding for individual and team tasks as opposed to the baseline condition of manual route planning. There were two experimental conditions: the Aided condition, where a Route Planning Agent (RPA) finds a least cost plan between the start and rendezvous
points for a given composition of force units; and the Baseline condition, where the commanders determine initial routes manually, and receive basic feedback about the route. We demonstrate that the Aided condition provides significantly better assistance for individual route planning and team-based re-planning.

According to “Interaction without Commitments: An Initial Approach” [14] it describes a number of theories of commitments which reply on the concept of agent belief, desires and intension (BDI). It shows that implicit commitments drivable from communications are sufficient & grantable in self organized agents. It shows that agent cannot be guaranteed of any social knowledge in completive environment. It describe the algorithm for finding deontic states.

According to “Search Algorithms in Intelligent Agents” [15] describes the search algorithms used in agents system. This paper will examine intelligent search algorithms that can be used to perform fast and reliable distributed searches. Some of the algorithms examined will be the Generate-and-Test, Best-First, A*, and Means-Ends searches. After reviewing some intelligent distributed search techniques, an example of how these techniques can be used within an emerging distributed system. They range from traditional search methods, to heuristically driven searches, through asynchronous dynamic programming methods. It shows comparison between them. It also defines time complexity of these algorithms.

According to “The RETSINA MAS, a Case Study” [16] identify challenges that confront the large-scale multi-agent system (LMAS) designer, and claim that these challenges can be successfully addressed by agent-based software engineering (ABSE), which we consider to be distinct from object-oriented software engineering for multi-agent systems (OOSE for MAS) in its consideration of agent goal, role, context and attitude as first class objects. We show how we have discovered these principles through our experiences in developing the RETSINA multi-agent system, in implementing specific test applications, and in the derivation of three distinct architectures that help guide and describe the designs of our systems: the individual agent architecture, the functional architecture, and the infrastructure architecture.
According to “HILBERT & PATRIC: Hybrid Intelligent Agent Technology for Teaching Context-Independent Reasoning” [17] describe that there is a disturbing paradox at the heart of contemporary American education: As this education turns more ‘electronic,’ we are moving away from the one kind of learning that we know to be most effective, namely, one-on-one instruction. As the need for good teachers at the university level continues to grow, we see this paradox intensifying. A specific exacerbating force is that, as the data tells us, as educators, we are not producing students able to successfully employ context-independent reasoning (CIDR) in technical domains. Our future scientists and engineers have been shown to lack this fundamental tool of their trade. The fact is, teaching students to be good CIDR reasoners requires the professor to develop a one-on-one relationship with each student, with a keen eye on how each applies their own strategies for proof construction. With this in mind, we envision the automated logic instructor as adaptable, and fully available to each student, at all times. This is obviously not possible with human instruction, but our preliminary work suggests that our vision is capable of being realized in the digital domain: We are developing a suite of intelligent agents, including PATRIC and HILBERT that marries the cutting edge in AI-driven tutoring with the state-of-the-art in proof construction courseware.

According to “Autonomous artificial intelligent agents” [18] reviews the current state of the art in the research concerning the development of autonomous artificial intelligent agents. First, the meaning of specific terms, like agency, automaticity, autonomy, embodiment, situatedness, and intelligence, are discussed in the context of this domain. The motivations for conducting research in this area are then exposed. Several principles that should guide autonomous agent research are reviewed. Of particular importance are the embodiment and situatedness of the agent, the principle of sensorimotor coordination, and the need for epigenetic development and learning capabilities. They ensure the adaptability, flexibility and robustness of the agent. Several design and evaluation considerations are then discussed.

According to “An Algorithmic Theory of Mobile Agents” [19] describes mobile agents are an extension of multiagent systems in which the agents are provided with the ability to move from node to node in a distributed system. While it has been shown that
mobility can be used to provide simple, efficient, fault-tolerant solutions to a number of problems in distributed computing, mobile agents have yet to become common in mainstream applications. One of the reasons for this may be the lack of an algorithmic theory which would provide a framework in which different approaches can be analyzed and the limits of mobile agent computing explored. In this paper we attempt to provide such an algorithmic theory.

According to “An Agent Based Intelligent Tutoring System For Parameter Passing In Java Programming” [20] describe an agent based intelligent tutoring system for the parameter passing mechanisms in computer science, an introductory Java programming language, in Al-Azhar University in Gaza. The agent based intelligent tutoring system helps students better understand parameter passing mechanisms in Java using problem based technique. In this paper, we will describe the architectural design and features of the agent based intelligent tutoring system. An initial evaluation of effectiveness of the system was carried out and the result was found to be positive. The evaluation confirmed the established hypothesis that using the intelligent tutoring system would result in an improvement in the learning of the students.

According to “An Integrated Token Based Algorithm for Scalable Coordination” [21] describe Efficient coordination among large numbers of heterogeneous agents promises to revolutionize the way in which some complex tasks, such as responding to urban disasters can be performed. However, state of the art coordination algorithms are not capable of achieving efficient and effective coordination when a team is very large. Building on recent successful token-based algorithms for task allocation and information sharing, we have developed an integrated and efficient approach to effective coordination of large scale teams. We use tokens to encapsulate anything that needs to be shared by the team, including information, tasks and resources. The tokens are efficiently routed through the team via the use of local decision theoretic models. Each token is used to improve the routing of other tokens leading to a dramatic performance improvement when the algorithms work together. It presents results from an implementation of this approach which demonstrates its ability to coordinate large teams.
According to “The Role of Intelligent Agents on Learner Performance” [22], the use of a male and a female intelligent agent in education is explored. Qualitative analysis of learner performance, motivation, availability and support of the agents, agent gender, and human-like characteristics is presented. Our intelligent agents converse via speech and text with students on matters concerning the design and development of their electronic portfolio. Our contributions to the current literature are twofold. First, unlike current practices, we examine the use of intelligent agents over multiple sessions. Second, the dialogue between agents and learners is not predetermined. This allows the learners to be active participants in the learning process and interact with the agents as if they were interacting with an instructor, expert, or peer.

According to “An Intelligent Agent for Adapting and Delivering Electronic Course Materials to Mobile Learners” [23], the concept of mobile learning is defined as learning that takes place with the help of mobile devices to access course materials; however, these devices operate in different ways and have different capabilities. To make course materials available on these mobile devices as well as on desktop systems, a software program is required to allow course material to be delivered in heterogeneous computing platforms. Because of these heterogeneous computing platforms, some course materials may not be in a format that different mobile devices would accept. It has developed an intelligent software agent capable of adapting to the heterogeneous mobile computing environment. The agent can search for a conversion tool according to the desired format and convert the course materials automatically. The agent is able to understand mobile clients’ capabilities. In order for the server to know what type of course material the client wishes to receive, the client needs to feed information on the software and hardware capabilities of the device to the server.

According to “The Agent Learning Pattern” [24], the development of large scale multi-agent systems (MASs) requires the introduction and structuring of the learning in agents throughout the design and implementation stages. In open systems and complex environments, agents have to reason and adapt through machine learning techniques in order to achieve their goals. This paper presents the Agent Learning Design
Pattern that guides the object-oriented design of machine learning algorithms in Software Agents.

According to “Agent Support for Planning Under Policy Constraints” [25] describe Mission-critical scenarios, such as military or disaster response missions, often call for the formation of coalitions, made up of people from different countries or organizations and required to adhere to certain policies. These policies define the explicit obligations, permissions and prohibitions governing members of the coalition. While planning for joint action in these scenarios is already a complex problem for human planners, it is made more difficult or even impossible under such policy constraints, especially if policy conflicts exist between them. It proposes that agents could be used to support human planners in coalitions, and present work in the area of agent support for coalition mission planning under such policy constraints. It defines a taxonomy of policies and outline the different types of support that agents can provide to human planners. It describes an experimental framework within which different types of agent support can be empirically evaluated within the context of a human planning problem.

According to “Domain-Specific Architecture for Software Agents” [26] describes methodology for the support of developing domain-specific software agents and the development of a Reference architecture for agents pertaining to web service discovery by following the phases described in the proposed methodology. The proposed methodology and the resulting architecture are evaluated to illustrate its appropriateness in contributing for domain-specific software architecture. This paper is designed to overcome the limitations of developing an agent from reusable architectural artifacts by:

- Proposing a methodology for building domain-specific architecture for software agents.
- Developing domain-specific architecture for web service discovery agent based on the proposed methodology.
An evaluation step is also carried out both for the proposed methodology and the developed domain-specific software architecture for software agents in web service discovery.

According to “Learning Objects for Intelligent Agents” [27] describes the details of some common characteristics of applications using Intelligent Agents as they relate to Learning Objects as software systems in delivering education. Readapting Learning Objects to different categories of learners constitutes a challenge for Intelligent Agents in their effort to provide a large scale of collaboration between different e-learning organizations. In order not only to have efficient access to Learning Objects, but also to offer to learners tutoring and mentoring help, collaborative and cooperative learning strategies, learning advancements, and social interactions, Intelligent Agents have been highly recommended by a number of researchers. This study investigates how these e-learning applications are designed, how students' differences are explored, and how these software systems are able to improve learning and teaching performances.

According to “Intelligent Agents: A Physics Education Opportunity in Latin-America” [28] describes Intelligent Agents are being applied in a wide range of processes and everyday applications. Their development is not new, in recent years they have had an increased attention and design in learning and as mentoring tools. It discuss the definition of what an intelligent agent is; how they are applied; how they look like; recent implementations of agents; agents as support in learning process; their state in Latin-American countries and future developments and trends that will permit a better communication between people and agents.

According to “The Research of Intelligent Negotiation Agent -Application for B2C E-commerce” [29] describe that B2C e-commerce is becoming more widespread as more people come to recognize its convenience and its ability to rapidly respond to requests and as more products and services become available. However, many electronic marketplaces, especially in the business-to-consumer, are in essence some kind of search engine where buyers look for the best product in a database of products offered by sellers. In an automated negotiation, intelligent agents engage in broadly similar processes to
achieve the same end. In more detail, the agents prepare bids for and evaluate offers on behalf of the parties they represent with the aim of obtaining the maximum benefit for their users. Nevertheless, in the current situation, price is the only criterion by which agents are created. This factor is easy to measure and automate. However, the criteria for advanced transactions need to be elaborated, for example, details of giveback and dividend. This paper presents a multiple-attributes negotiation model for B2C e-commerce, which deploys intelligent agents to facilitate autonomous and automatic online buying and selling by intelligent agents while quickly responding to consumers.

According to “Educational Simulator of Conic Sections Supported by Intelligent Agents and Semantic Web: An Example in the Development and Design of Educational Simulators for Mathematics” [30] describes the main aspects in design and development of an educational software tool that belongs to the simulators category, proposed to support teaching and learning processes in the topic of drawing conic sections, such as the hyperbola, parabola, circumference and ellipse in rectangular coordinates in mathematics field. The educational simulator was designed using a multiagent system with ontology in order to increase the application performance and the feedback to students in web learning environments.