Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Testbed For Intelligent Agent: A survey

[1]Ahmed M. Kareem, **[2]Prof. Dr. Ali Obied,**

(1) College of Computer Science and and Information Technology , Al_Qadisiya University, Diwaniyah,Iraq, Email: com.post08@qu.edu.iq,
(2) College of Computer Science and and Information Technology , Al_Qadisiya University Diwaniyah,Iraq, Email ali.obiad@qu.edu.iq.

ARTICLE INFO

Article history:
Received: 15 /05/2021
Rrevised form: 28 /05/2021
Accepted : 24 /06/2021
Available online: 07 /07/2021

Keywords:

Benchmark and Testbed, Intelligent Agent, Classical Planning, TileWorld, PHOENIX.

ABSTRACT

Revealing the features of the intelligent agent and its interactive behavior with complex environments whose events cannot be expected is a source of interest for researchers. What we will put in the hands of the interested person is to review key concepts in the design and implementation of test environments, noting the importance of the testbed, in addition to the knowledge bases required by these designs, presenting the definition of the intelligent agent and its features to be disclosed within the testbed. As for the types of test beds and their features, they have an important aspect in these papers.

MSC. 41A25; 41A35; 41A36

DOI : https://doi.org/10.29304/jqcm.2021.13.2.816

1.Introduction

In recent times, controlled experiments within AI research are in a significantly increased state, with changes to the system or environment by designers in the specifications of that system that affect the aspects of the system in terms of efficiency. Two vocabulary that are well-known in the community of researchers[1][2], namely:

1) Benchmarks that are units of measurement for key concepts.
2) Testbeds (virtual environments)

∗Corresponding author: AHMED A. BRISAM.

Email addresses: Ahmed.brisam@qu.edu.iq.

Communicated by: *Dr. Rana Jumaa Surayh aljanabi.*

There are some challenges related to the appropriate use of the experimental method, and this does not reduce our impulse that this approach has support for artificial intelligence.

1) The outputs are not guaranteed from the testbeds and the benchmarks, as it is a part of experimental artificial intelligence. Noting that it is the responsibility of researchers to differentiate between important and unimportant phenomena[2].
2) Scientific deception: the case of targeting by investors (program investors) and those who provide strong support, this puts us in the problem of lack of confidence in the employment of test beds and benchmarks, as they are not used as accurately as intended, a case like this gives us an attractive scientific illusion.

Two main goals behind the benchmarks and the testbed, the first of which is to set clear indicators (matrices) through which we can choose between systems that have similar specifications (competing), at the same time we need a theoretical coverage of the practical effort[2]. The scientific competence of the testbeds and the benchmarks architectural measures lies in their ability to uncover the scientifically agreed upon interesting aspects to evaluate the performance of the system.

We think the practical (experimental) control achieved by employing the test bed is considered a distinct assistant in the perception and interpretation of the reasons for the behavior of the systems under examination and testing. Artificial intelligence systems are distributed in different locations with highly complex, interfering and complex environmental influences, the most important aspect of the test environment is the simulation that has a literature that must be adhered to when designing the testbed, and achieving a high level of reproduction status for the realistic environment. The controller of the virtual environment has the right to change the parameters (features) of the environment and calculate the effects on the system that is under test, provided that these variables are random and not random. There is a very important question, this question stems from the difference between realism and the ability to control practical (experimental), this difference can be reduced to an acceptable degree when testing systems that operate within an environment described as deterministic, and this is what we will address in the environmental characteristics.

In other words, the focus of our attention does not revolve around interacting with simple systems within uncomplicated environments, the fact that artificial intelligence systems distributed around the globe must deal with extremely complex environments, and we do not close our eyes to simple designs, which provides us with benefit when we simplify Complex systems into multiple simple form combined in a system described as complex.One of the best solutions is to pay attention to designing an environment close to reality and to give satisfactory and clear results for systematic experiments directly on those environments and there are methods that must be addressed. The acceleration of interest in testing beds, Benchmarks and experiments subject to artificial intelligence, such as understanding natural languages and machine learning, what we want to focus on is its clear impact on the design of the agent, including the features contained in the test bed that help us highlight the attractive features of the agent and the corresponding design lapses or weaknesses affecting the performance of the agent, discussing some of the questions and inquiries that increase the level of argumentation, allowing us to talk together.

2.Benchmarks and Test Beds

Benchmarks have a clear aspect in computer science as a tool. An example illustrates this. Matrix multiplication is a tool in computer science, in other words, a good Benchmark. The designers of the computer's CPU do not tolerate in focusing on the speed of the processing unit, so the use of matrix multiplication to give them an image of the speed, then matrix multiplication as a tool is a criterion for evaluating the serious performance of the processing unit. An Sussman anomaly was an early Benchmarks for AI planning programs (the three-block problem)[3]. Sussman anomaly Support many researchers in uncovering the mechanism of their schematic work, and the reason for fame simulates multiplication matrices with what it represents of an important class of problems that have interactions and interrelationships between the branching objectives. It is easy to describe.

Let's first ask what is the behavior of the agent? The behavior of the agent is according to the context around it, as it is aware of what is happening in order to adopt a specific action, The agent who can adapt to the surrounding circumstances, it is the intelligent agent, It has characteristics and features with details that we will mention in the subsequent sections. Benchmarks draw us the behavior of the agent at a bright level, this agent that we are interested in, and usually focus on part of the behavior of the agent and not all the behaviors, so testing all behaviors means putting the agent in several tests, in other words, adopting a number of Benchmarks, whether you are a scientist, a designer, or As a consumer, this dictates what you want to know about the agent's behavior. In instance, a consumer of search algorithms within artificial intelligence, this consumer may appear excited by what he sees from the outputs of the algorithm for min-conflicts heuristic in the problem of the n-queens in time is constant [4]. While in roles different from that of the consumer, such as scientists and designers, their aims is to realize the reasons behind the behavior, what is the feature that the algorithm enjoys so that it performs its tasks at high speed? Why is the problem of growing n-queens problem solved in an almost constant time when using the min- conflicts heuristic algorithm?

Setting Benchmarks as a tool for detecting the behavior of the program needs modeling, as this modeling achieves the goal of scientists and designers with a deep vision and an answer to their inquiries. We are taking a risk when we do not adopt modeling, without modeling, the program may pass the testing phase without having indicators to pass the test. Benchmarks are difficult with a model of parameters, and this difficulty is required for the success of the Benchmarks in design. The model describes to us why the program performs well, the experience using the Benchmarks gives us design options with excellent performance. The use of a problem to test the performance of a system is not sufficient for researchers, for example employing the n-queens to test algorithms such as this test does not give a timeline of events. Benchmarks in a clear format are problems for each system, as it has its own tools to solve, and it is not specific to the details of a particular system. It is a source of learning for what the researcher wants to know about the system and its performance, which is inversely proportional in terms of complexity to the system and its properties. The researcher has the option to evaluate the behavior of the system or to evaluate parts of the system.

3. Issues in Agent Design

Despite the difficulties in designing a testbed, as that environment has a close relationship to the experiment or its almost total dependence on experience in evaluation, there is a development witnessed at the present time for these environments in studying the design of the agent and investigating some of his behaviors, where the agent effects on his environment [5]. In this section, we will discuss the definition of the agent and his of behaviors on the experimental environment, which it revealed, the most important of which is the extent of his ability to adapt to his environment and make changes in it.

3.1 Agent Definition

From artificial intelligence and distributed programming came smart agents. The two sciences have united to appear smart customers, and it is called the science of distributed artificial intelligence[6]. The idea of the intelligent agent in 1950, and nothing actually came into being until the end of the seventies. The topic was widely known in the early 1990s, along with the Internet[7]. There is an analogy between an agent and a directed object being distributed programming and artificial intelligence [35]. Researchers speak of artificial intelligence, not human, when referring to the intelligent agent as it deals with devices such as computers [8]. The human have a different behavior [9]

The agent did not meet the definition agreed upon by the researchers. We will provide definitions used by researchers, and we will set some characteristics for the agent, ending this field with a definition of the agent from our point of view. The intelligent agent is a difference from the intelligence control system, intelligence has as a measure of the various goals that are within the agent's endeavors to achieve them, the multiplicity of positions to achieve those goals, and the no simple relationship between the ends and the means upon which the quest is built.

3.2 Agent characteristics

The different opinions about the interpretation of some of the characteristics (what means autonomous) by researchers, one of the reasons that gave various definitions of the intelligent agent. What we have chosen from the characteristics is the most important focus of researchers, directly or indirectly, to define the intelligent agent. However, these characteristics are not the only ones that all researchers have adopted.

3.2.1 Autonomy and Intelligence

Carrying out autonomous actions is the most important thing agreed upon by researchers. An autonomous agent has the ability to control his actions and behavior, it acts without the interference of others (user or agents), in addition to his the initiative in achieving his goals[10]. Autonomous and intelligence some scholars equate them. Meaning, they consider the ability to take initiative on the part of the agent an advantage for being an intelligent agent. Foner believes Julia is an intelligent because it is autonomous and has the initiative[11]. But the researchers are not talking about human intelligence, but machine intelligence[8]. That is why Petrie spoke of autonomous and not about intelligence, because he links the word intelligence with human intelligence, which is difficult to describe.

3.2.2 Learning

Intelligence is a word that has the quality of learning. The question arises for the agent to be intelligent, does it have to be learning? A number of researchers believe that learning is an important attribute of an agent[11], while others never mention it[10].
When an agent is present in a dynamic environment, it is not possible to know what it will face[12], so the agent is distinguished from others when it has the ability to adapt and learn from his environment. What is a learning program? It is a program that uses memory to solve problems after saving things. Learn the agent from monitoring the user, its environment, or other agents. The agent is in contact with others (user or agent) by means of natural symbols or language that guarantee its communication, and it is a two-way letter. The second way to learn is from the agent's own programming[13]. Thus, the methods of machine learning are either by communicating or by programming the agent itself.

3.2.3 Co-operation

Complex problems may require more than one agent to cooperate in solving them[7]. Cooperating agents agree on the goals to be reached and achieved. When using agents, cooperation is the main distinguishing part of them[14].
Foner deals with problem solving in cooperation between agent and user[11]. The cooperation is in two dimensions, it may be between the user and the agent, and it may be between the agent and other agents. Agents 'collaboration breaks tasks down into smaller, distributed problems for agents to solve. This cooperation takes place in the presence of communication between the assistants.
If we imagine that there is no communication, then how can cooperation be? The answer is through observing. The agent is the one who determines whether or not the second party needs it. This is a manner work without communication[7].

3.2.4 Lifelike

Agents lifelike, it is the style of delusions, so emotions and social interactions are within the capacity of the agent. Various techniques and mechanisms are required to create lifelike characters, including speech recognition and animation. In addition to psychological sociology and methods of dialogue, it is necessary to understand them [7]. It is interesting to be agents with human characteristics, such as emotion, that can be portrayed as a realistic human face[15]. Some researchers argue that showing realistic traits through an agent is an imperative[16]. A spontaneous agent, not a mechanical one, behaving unpredictably, it is a lifelike agent.

3.2.5 Mobility

It is the movement through the network, such as the Internet. Is the agent capable of this movement. Researchers believe that moving the agent across the network provides a new method for the process of retrieving data and transactions[17]. The agent can move from one server to another to find the best journey for you, and give you the result by that. The disadvantage in this matter is that the agent may present to your personal computer many options that consume time and oblige you to stay connected to the Internet.

3.3 Summarized Definition of Intelligent Agent

The definition of the intelligent agent suffered a lot of divergence in views from the researchers, we hope that we have a clear vision that is useful in defining the intelligent agent, which gives a simplified and comprehensive knowledge base on this concept.
The main idea in defining the agent, which does not contradict the opinions of the researchers[7][18][14][19], is an entity with a purpose, there is a motive to build it in assisting the user in a specific task and in a specific environment, for this intelligent agent has basic features (autonomous and learning). autonomous is a feature that distinguishes it from other programs. The autonomous agent is in addition to the previous definitions of the meaning of the autonomous, such as his ability to self-act and make decisions. The agent has an internal encapsulated state.
When the environment is dynamic, the agent must be aware of changes, it has knowledge that enables it to carry out its task. The learning feature is a human feature that enables us to be smart. Communication and cooperation, they are not two main qualities, we have known their role, but the reason behind making them non-principal is the possibility of designing an intelligent agent without them, for example an agent may learn through communication, but it is possible to learn from monitoring and this simplifies the internal structure of the agent, as for cooperation between agents it is to simplify the complex task, break it down and distribute it to other agents. We agree with Maes that the advantage of a lifelike agent is spontaneity and unpredictability[16]. The entity's behavior spontaneously provides us with information that we may be interested in and did not request. As for mobility, it has an effective role in migrating the agent to another location to carry out the task.
Communication, Collaboration, lifelike, Mobility, Four typical characteristics of an agent, which he is likely to have all or part of. these characteristics do not preclude the agent may has additional features, such as reliability. This understanding of the intelligent agent takes us to Hayes Roth, who says that the domain in which the agent works must be defined[20]. where the field determines the features that the agent should have, and we will not end up with an agent that is too complex for the task entrusted to it. A simple agent may be the best solution. Tool box, yes, agent characteristics, a box in which a set of tools for us to choose what is necessary to solve the problem, with the two main characteristics being autonomy and learning. The distinction between human intelligence and machine intelligence is a concept that must be understood. Intelligence is automatic for the agent because it does not exercise human intelligence, so we see the intelligent agent falls within the framework of its ability to autonomy and learn, or we name agents according to the purpose for which they are designed, such as information agents Nwana[15]. Figure 6 shows what we're adopting.

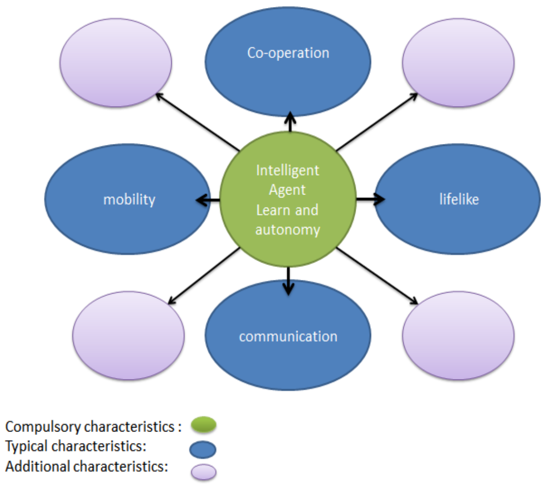

Figure 1 : Intelligent Agent Characteristics

3.4 Agent architectures

Defining the intelligent agent in advance makes it easy for us to understand the various structures within an agent design. What we mention here is a number of structures based on the action of the agent, i.e. the mechanism of interaction of the subject with his environment. The main point in this part is what is the design of an agent that achieves the previous characteristics such as autonomy, reactiveness, proactive, social ability and so on.

3.4.1 Purely reactive agents

The agent who relies on the mechanism of choosing its acting without relying on its awareness of previous information Therefore, the actions of the agent are selected based on the current state (the agent's perception of current information). We have in this type of agents a direct response to its environment. It's may be called Simple reflex agents. A simple example of a purely reactive agent, the water level agent in dams, it controls the water level at a certain level in the dam, closes and opens the gate, as well as the thermostat is an example of a purely interactive agent.

3.4.2 Utility-based agents

an agent have set of accessible runs. Each run has a bunch of activities to execute a goal. A utility means a numeric assessment on how perfect a particular executing is given the current awareness information from the world. A type of utility based agent, including a function that calculates the utility as a real number dependent on information from the perception of the environment.
With a high degree of utility, the agent accomplishes the operation process. This method ensures that there are attempts by the agent to improve its performance to the maximum extent. In most environments, to generate high quality behavior we do not rely solely on goals. For example, a series of procedures achieves the goal by the arrival of the car, but part of those procedures are not safer or less expensive. The comparison of the happy state with the unhappy is done by means of a measure of performance. The term happiness has been replaced by the term utility(it's a term scientific). The meaning of utility is the quality of benefit. In this way, it is easy to distinguish between procedures that are more desirable. A utility function of agent is basically a capacity of the measure of performance. when there is an agreement between the external measure of performance and the internal function of utility, choosing action's high utility by agent will be rational with the external measure of performance[21] . The utility-based agent makes decisions based on utility, although the goals are insufficient, when the goals are insufficient, we have two situations: The first is the goals are conflicting (such as speed and safety), here the utility is on the barter (appropriate tradeoff). Secondly, there are a number of goals, but they are not achieved with certainty. The benefit here depends on the probability of success versus the importance of the goals. We should not fail to mention that our partial observable and stochastic environments are widespread in the real world. A rational agent makes decisions based on utility, this is the utility the agent expects in those environments.

3.4.3 Goal based agents

The agent relies on an action plan to achieve the goal. The process of selecting a plan is not simple, the agent uses research and planning processes. Depending on the goal and perception, plans are drawn up, after which the agent checks if the goal can be achieved according to perception. In addition, the previous information may be used to define the procedures (selection process) and to identify what needs to be done. For example, awareness is not sufficient to determine the necessary measures, in traffic intersections, the car has to turn left and right, or it is moving straight. Making the right decision here depends on the driver's target location and the rider's desire to have an influence on decision-making. When an agent has to contemplate long, fluctuating steps and turns in order to find a line to achieve the goal, it is much more difficult. Therefore, research and planning are partial areas of artificial intelligence that involve finding and adopting a series of measures that achieve the goal[21].
The agent has a prospective study of the effects of its actions, and no single procedure is adopted. For example, when the driver sees a red lamp, he does not relay one action only, which is to press the brakes directly, but rather takes several procedures, its decisions are more flexible as they can be adjusted. The agents and how make decision can be brief by Table 1.

3.4.3.1 Rational agents

Wooldridge and Jennings provide a list of the abilities expected to be contained in the design of a rational agent[22]:
Reactivity ability: Perception of the environment is among the capabilities of the rational agent, and in a timely manner it has a response to the changes occurring in the environment, this in response to the design of capabilities of the rational agent. Proactiveness ability: Taking the initiative, adopting goal-oriented behavior by initiative, in response to the design of capacity of the rational agent. Social ability: the ability to communicate and interact with other agents, in response to the design of capabilities of the rational agent.
Doing something is considered right, what does that mean? It is a question that can be answered, but our answer will be within a framework that interests us, which is the behavior of the agent. The agent is immersed in the environment, from which a set of procedures is issued according to his perception of that environment, this group or sequence of procedures has consequences for the environment and it is a sequence of states. If the sequence of states in the environment is desirable, we say that the agent has a good performance, that desirability or approbation for the performance of the agent is captured through performance scale.
The focusing of the evaluation process is on the states of the environment and not on those of the agent. Noting that the factors for measuring the efficiency of the agent are varied according to the purpose for which the agent was designed. To clarify the idea of relying on environmental conditions(states) in the evaluation, for example the vacuum cleaner, if we rely on the agent's states in the evaluation, it will pick up the dirt and record a point for its, then throw it and pick it up again, also a point is calculated for him, while when the evaluation is on the cleanliness of the room after dividing it into Squares, and every clean square takes a point, so the scale is according

to what we want and not according to what the agent wants. From The foregoing rationality has three pillars, the first is the measure of performance, which determines the level of achievement and evaluates it. Secondly, prior knowledge of the environment by the agent, in addition to the agent's awareness of the sequence of states in the environment, third and last possible actions of the agent[21].

Agent type	How take decision	example
Purely interactive agent	current state	Thermostat
Utility-based agents	utility	route recommendation system(RRS)
Goal based agents	Research, planning, and perception	Procedural Reasoning System(PRS)
Hybrid agents	reactive and proactive behavior	Cooperative Intelligent Real-Time Control Architecture (CIRCA),

Table 1: Agents type with their decision

3.4.3.2 Definition of a rational agent

For every conceivable percept sequence, a rational agent should choose an activity that is relied upon to boost its measure of performance, given the proof given by the percept succession and whatever underlying information the agent has[21]. Examples using rational term, BDI AGETN (Belief- Desire- Intension), PRS (Procedural Reasoning System), and IRMA (Intelligent Resource Bounded Machine Architecture).

3.4.4 Hybrid agents

For an agent's reactive and proactive behavior in the same structure, it's problem, the solve that is introduction of hierarchy of interacting sub-system layers, this is the well- used option[23]. In hybrid architecture, at least two horizontal layering, one of them to agent's reactive behavior and other to agent's proactive behavior, but stay issue in this structure, how agent control to take decision between the two layer or who is layer take the control option.

All layers in a horizontal layering architecture are sharing in the perception (input) and the action (output), a comeback on the start, must there is a control function that is considered a bottleneck in the agent's decision making. this function decides any one of the layer has the control of agent. See Figure 2 [23].

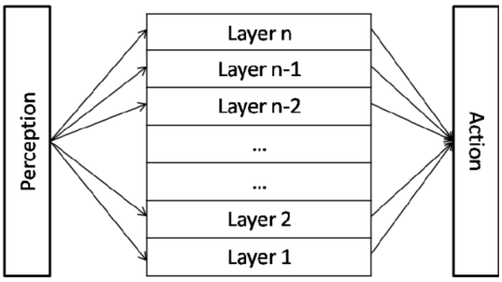

Figure 2 : Horizontal architecture

In the second type of layered architecture, that is vertical layering, see Figure 3 [23]. At least one layer interaction with sensor and action(input and output), a vertical layering has two kind the first is one-pass and the second is two pass architecture. Through each layer and in a sequentially direction the control and information streams, the last layer creates an action output, that is described one-pass architecture. Otherwise, in two-pass architecture, the shape of the control and information streams are in two directions, flows up and flows down through the layers.

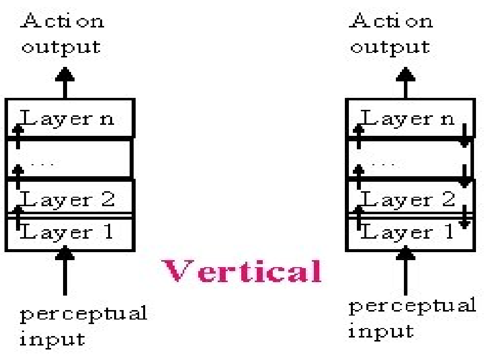

Figure 3: Vertical architecture

4.Environment characteristic

The real world around us has a lot of constants and variables that we deal with on a daily basis, this is what triggered the researchers thought to develop an environment with specifications that mimic our real science, describing the properties of the environment. We can simulate the agent in the real environment. Russel and Norvig have put together a description of the agent's environment[21].

4.1 Fully observable vs. partially observable:

If the agent is able at every moment through its sensors to access the entire state of the world, then the environment task is described as the fully observation. When the sensors are able to detect all aspects related to the filtering of action, the environment task is fully observed. The agent in the fully observable environment does not need a history to keep the state of the world in order to track it.
In the state that the sensors suffer from deficiencies for design or external reasons, the data is incomplete about the world, this environment is unobservable, like the driver of the autonomous car cannot predict what other drivers are thinking, but despite this situation, which seems hopeless, however, the goals of the agent achievability remains possible.

4.2 Single agent vs. multi-agent

The distinction between single-agent and multi-agent environments may seem simple enough. For instance, a crossword puzzle game is an environment with a single agent responsible for finding the solution, while chess is a two-agent environment. In a single agent environment there is one agent operating whereas in multi-agent environments there are many agents that interact with each other, but at times objects or entities that we would not normally consider as agents may have to be modeled as such. Nature may be modeled as an agent usually any entity/object that affects or influences the behavior of the agent under consideration needs to be regarded as an agent.

4.3 Deterministic vs. stochastic

When the agent can depended on the current state to know the next state of the environment in which it embedded, and the action is carried out by the agent then we can describe this environment as deterministic, otherwise it is described as stochastic. This is our talk when there is one agent. In cases characterized by more than one agent, it is deterministic when each agent can predict the actions of other agents. Thus, the partial observation environment classified as stochastic environment. Most aspects in the real world are difficult to fully observe, which is why it is treated as a stochastic environment. For example, a self-driving car has a stochastic world, so the agent cannot fully predict the traffic, in addition to design events such as engine failure without warning. From the above, we come out with a conclusion: the uncertain environment is one that is not fully observable (nondeterministic). When we have no certainty about the results, the stochastic environment has no probabilities associated with the results.

4.4 Episodic vs. sequential

In episodic state of the world, the experience of agent is partitioned into molecular episodes. The agent is gave an data and then executions an one action, that is in every episode. Critically, the following episode doesn't rely upon the activities past episodes are taken. The episodic is classing for several tasks. For example, the agent makes a decision regarding the current part without being indifferent to the previous decisions in determining the defective parts on the production line, in addition to that, if the next part is defective and not affected by the current decision. On the other hand, in a sequential world, future decisions are affected by the current decision. Autonomous cars and chess are both sequential worlds, in which long-range convoys can be caused by actions of short range. The sequential world is much more complex than the episodic world, because the responsibility of the agent in the sequential world is a think ahead.

4.5 Static versus dynamic

a static world, the agent does not need to look at the world without interruption while deciding on a specific action, in addition to the fact that the passage of time does not pose a concern to the agent, so it is easy to deal with static environments. When an agent is in a deliberation and changes in the environment may occur, that environment is termed the dynamic of that agent. The dynamic environment continuously asks the agent a question, what do you intend to do, if a decision has not been made yet, this is a decision to do nothing. We have another name, which is semi-dynamic, this name refers to the environment that changes due to the action of the agent. If we put our previous examples under the microscope, for example, crossword puzzles, a static environment, chess game, a semi-dynamic environment, and a self-driving car, its environment is dynamic, because other cars are in motion at a time when the driving algorithm has not yet determined what to do.

4.6 Discrete vs. continuous

The discrete/continuous differentiation applies to the condition of the world, to the manner in which time is dealt with, and to the agent's perceptions and activities. For instance, a limited number of distinct states in the chess world (barring the clock). Also a discrete set of perceptions and actions are in chess world. while, there are state and time continuous problem in self-taxi driving : the taxi's position and speed and other cars move through a scope of continuous values and do so easily with time. Self-Taxi-driving activities are likewise ceaseless (continuous) and the input by set of cameras is discrete.

4.7 Known vs. unknown

Strictly speaking, this description is not directed at the environment per se, but rather at the agent's state of knowledge about its environment. All results are known for all actions, this is in the known environment. In an unknown environment, the agent must learn how that environment works in order to be able to make the appropriate decision , and agent have mechanism how make decision[24]. Note, the difference between the full observation and the partial observation in the environment is not the same as the difference between the known and the unknown environment. For example, in solitaire games we have knowledge of the rules of the game, but there is no vision for the cards that have not been received, it is a known environment and partial observation environment. And an example reflects the other picture when the environment is unknown and fully observed, when watching a new game video, we have a complete vision of the game but we do not know how to use the buttons.

As one might expect, the hardest case is partially observable, multi-agent, stochastic, sequential, dynamic, continuous, and unknown. Taxi driving is hard in all these senses, except that for the most part the driver's environment is known. Driving a rented car in a new country with unfamiliar geography and traffic laws is a lot more exciting. Table 2 lists the properties of a number of familiar environments. We have not included a "known/unknown" column because, as explained earlier, this is not strictly a property of the environment. For some environments, such as chess and poker, it is quite easy to supply the agent with full knowledge of the rules, but it is nonetheless interesting to consider how an agent might learn to play these games without such knowledge. As expected, the most difficult case is (sequential, stochastic, multi-agent, partially observable, dynamic, continuous, and unknown). The self-driving car has a distinct environment with all the above difficulties other than unknown, being available to it. In the table below, we review common environments and offset their characteristics, and we did not include known as it relates to the agent as mentioned earlier. The interesting question about how the agent learns and provides it with knowledge, and this is under the heading reinforcement.

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	discrete

Table 2: (example of environments and their characteristics)

5.Classical Planning Paradigm

The classical planning paradigm accepts a climate that is both controlled and basic. The planning agent is for the most part accepted to have unlimited authority over the environment, which implies that its proposed activities are the lone events that can change the world's state and, besides, that the impacts of its activities are completely known, both to the specialist and to the framework fashioner.
The agent is generally expected to have total what's more, mistake free data about the condition of the world when it starts planning. Since it understands what the underlying state of the world is, the thing that activities it means to do, and what the impacts of those activities will be, it can at any rate on a basic level, anticipate precisely what the state of the world will be the point at which it completes the process of acting. At the end , it knows early whether a specific planning will or won't accomplish its objective.
Classical planners encapsulate solid working on suspicions both as in their abilities (the class of issues they can tackle) will in general be restricted and as on the planets wherein they work will in general be little, showing not many highlights and a restricted physical science. planner are for the most part tried in spaces with not many planning administrators, on objectives with few conjuncts, and on models of the world in which not many highlights are unequivocally displayed. Execution will in general corrupt when the quantity of administrators, objective conjuncts, or ecological highlights increments.
Similarly as control implies that the planner can, on a basic level, demonstrate that its arrangement will work, the improving on suppositions imply that the organizer can as a useful matter produce the verification. Control and improving on presumptions, along these lines, permit the planner the advantage of creating provably right plans preceding execution time. Most current work on agent designs points toward loosening up these presumptions. re-acting systems, for instance, manage the issue that the world can change erratically between plan time and execution time by choosing what to do at execution time as opposed to creating an arrangement preceding execution. Case-based planners defy the effortlessness issue by putting away just the fundamental subtleties of an answer, permitting the planner to focus on the pertinent highlights of another issue. Next we portray some particular issues that have as of late pulled in the consideration of preparation analysts and, subsequently, control choices about what includes an arranging proving ground may show.
Exogenous events: Perhaps the most restricting supposition of the classical planning worlds (most strikingly, the blocks world) is that no exogenous, or non-planned, events may be happen. Loosening up this supposition makes the way toward foreseeing the plans impact more troublesome [25] also, besides familiarizes the need with react to unconstrained events as they happen at execution time[26]. The time consumption of planning gets significant in a world that permits impromptu changes: The more extended the agent takes to plan, the more probable it is that the world has changed altogether between the plan's time was produced and the executed time the plan.[27][28][29]. The complexities of the real world: real world has its own characteristics, for example a simple block has color, shape, size, dimensions, etc., the real world has a complex interconnected structure, so change is one part of this world that changes other aspects. Thinking about a model of the real world we lack the ability to represent and predict about complex mechanics[30]. The test bed to simulate reality has a complexity and variety of features that enable it to detect complex planning problems. The cost and quality of observation and acting: observation and acting, generally neglected by classical planners, are not free. The agent should pay attention to the observation procedures in order to raise the level of informatics according to which the acting is[31]. Therefore, the testbed studies the design of the agent through models that illustrate this. The test bed needs an interface that distinguishes between agent performance, the simulation world, and those design revealing influences.

Proportions of plan quality: Classical planners are given an objective state to accomplish, and they stop when their arrangements can accomplish this state. Nonetheless, straightforward accomplishment of an objective state is a deficient proportion of achievement; It doesn't consider the expense of accomplishing the objective, and it additionally doesn't concede the chance of fractional objective fulfillment.
the connection between goal articulations and utility functions investigated by [32][33]. A testbed for investigating more extravagant thoughts of progress and disappointment ought to permit the architect to present issues including incomplete fulfillment of wanted states, driving the planner to exchange the advantages of accomplishing the intention against the expense of accomplishing it. The issue of adjusting cost against arrangement quality turns out to be more troublesome when the agent is really anticipating a succession of issues over the long haul, some of which probably won't have been made express when it starts to plan. Numerous agents: presents new issues when allowing various agents act in the world :
how practices are facilitated, how the agents ought to connected, how the impacts of synchronous activities vary from the impacts of those activities performed sequentially. Numerous agents planning is a functioning examination region [34], and a testbed for investigating these exploration issues should permit facilitated conduct and correspondence among the agents that occupy it.
One other thing with the capacities needed to make the proving testbed, we likewise distinguish some structure gives that will in general make a testbed more valuable to fitting users. A spotless interface: It is essential to keep an unmistakable differentiation between the agent and the world in which the agent is acting. The common partition is through the agent's observations and acting, so the interface ought to be perfect, all around characterized, and very much recorded. A designer should have the option to decide effectively what intentions are accessible to the agent, how the intentions are run by the testbed, and how the world's data is imparted back to the agent.
A characterized time's model: Test beds should introduce a sensible model of taking a spent time to reenact exogenous events and concurrent activity and to characterize plainly the spent time of thinking and acting.(This issue is an overall one in reenactment and modeling. See [35], for instance). Yet, the testbed should by one way or another have the option to match what amount time is passed in realistic world. Figuring out exploratory outcomes requires an approach to accommodate the proving ground's proportion of time with that utilized by the agent. Supporting experimentation: Varied problem and conditions model for evaluating agent architecture performance This is what the test stands for. Thus, enabling the experimenter to make changes in the environment in which the agent is tested is an issue of the test bed. There is the possibility provided by the test environment to monitor the behavior of the agent by the experimenter[36]. In addition, there are statistics used for evaluation[37], and this must be taken into account and allowed through the test bed and the emergence of data and analytical statistics in a clear and possible way to benefit from it.

6. Test-Bed Implementations

We previously reviewed the most important motives for building a simulation world (test bed) and the challenges we face in it. Now we present a review of environments committed to simulation literature describing those worlds in terms of design and use, in addition to summarizing the methodological foundations related to the test bed.

6.1 Grid Worlds

Test beds designed around the task of moving tiles within a two-dimensional network. This task is assigned to the agent. We'll look at three of the 2D test beds:
FIRST : the TILEWORLD of Pollack and Ringuette (1990) [38].
SECOND: the independently developed NASA (National Aeronautics and Space Administration) NASA TILEWORLD (NTW) [39].
THIRD: the MICE simulator [40]. Presented by Marenostrum Institut de Ciencies de l'Espai ` (MICE) collaboration

6.1.1 TILEWORLD test bed

Pollack and Ringuette (1990) report on the TILEWORLD test bed, an unexpected dynamic system that provides controlled, observable, and traceable experiences built for adaptive agent architecture. This environment is an incubator for the agent, tiles, obstacles and holes in its two-dimensional grid profile. The capabilities of the agent available within this system move vertically and horizontally, taking into account that the agent does not collide with one of the obstacles or the limits of the network, as for the agent's task is to close the holes with tiles. The holes have both capacity(C) and value(score). For the agent to push the tiles next to the gap towards it to close, the agent will have a rating of score(S). Every experiment has a period limit, agent performance is calculated by trial's score when time expires.
External events exist in TileWorld, objects (holes, obstacles, tiles) appear and disappear over time to give a simulation of the realistic world. The rate at which objects appear and disappear is under the control of the experimenter, in addition to capacity and score, which are properties of objects. The most important feature of TileWorld is the ability to control these parameters in a way that allows us to explore the different characteristics of the organized world (a world that changes relatively or slowly). Finding the systematic relationship between world features and those of the embedded agent is the overarching goal of this exploration, (see Table 3). The TileWorld framework is distributed with a fundamental agent architecture, which is likewise defined to permit control by the experimenter (see the accompanying conversation).
The agent is allowed to move up, down, and on both sides within an interface between him and the environment, for example when there is an obstacle blocking the road, so movement is not possible at that time for that site. The actions of an agent have predetermined and deterministic effects: the agent does not move to the adjacent cell if it is inappropriate and or lacks mobility, its movement is subject to calculations and is not subject to chance. Tiles and Obstacles only occupy one cell each and have their own network location and type. While holes have properties (capacity, location, and score) and occupy one or more cells. Probability is the controller of the appearance and disappearance of objects (holes, tiles, and obstacles) according to parameters set before the experiment by the researcher. The probabilities are autonomous of each other; a solitary likelihood administers the presence of tiles, and it is a similar paying little heed to the time, the area, or some other boundary in the game. TILEWORLD has no explicit sensing operators. The agent is provided with a data structure that describes the world's state in complete detail and with complete accuracy. The use of this information is left to the designer of the embedded agent; for example, he or she can design mechanisms that distort the information to introduce inaccuracy. The researcher describes a world by specifying the size of the grid; the duration of the game; and the probability parameters governing the appearance and disappearance rates of tiles, obstacles, and holes and the distribution of hole scores and capacities. The experimenter can control additional environmental characteristics; for example, the experimenter can decide whether hole scores remain constant until the hole disappears or whether the score decreases over time. TileWorld does not enable the agent to switch on or use the sensors, but rather gives the agent an accurate and complete view of it (full observation). It becomes up to the agent or embedded agent designer how the data is used. The world will be described in terms of grid size, available testing time, and probability parameters controlling tiles, obstacles and holes in terms of appearance, scores distribution and holes capacity. Another advantage of this world is by opening the way for the experimenter to control some world characteristics, for example, for him to decide whether or not to decrease the degrees of the hole with the passage of time. To easy experimentation, the world gives systems to indicating components of investigations, which would then be able to be run without mediation, and recording execution information.
Attributes inherent to TileWorld, which are simplicity in parameters, and what we mean by simplicity is the presence of a few parameters that enable us to define the outside world, reflecting to us the realism and simulation in the occurrence of objects as well as the ability of the experimenter to control those parameters. TileWorld isn't an endeavor to show a specific modeling area; in contrast, the world may be utilized to present paradigmatic planning issues in the theoretical.
IRMA is concerned with developing the TileWorld and in fact the TileWorld is included IRMA when distributing it to the research community. There are a number of agent architectures defined by IRMA, in other words, there are architectures for a group of agents within the IRMA concept and framework. The agent embedded step in the following allows adding a parameter to changes and options in the design in line with the IRMA framework. Working between the embedded agent and its environment within a clear interface is as follows:

- The occurrence of a desire by the agent to perform a work.
- The Simulator is called by the agent as a subroutine.
- The agent performs his work in the environment.

After the deadline of the agent to complete his work, the simulator carries out the external events and sends complete data of what happened in the world to the agent.

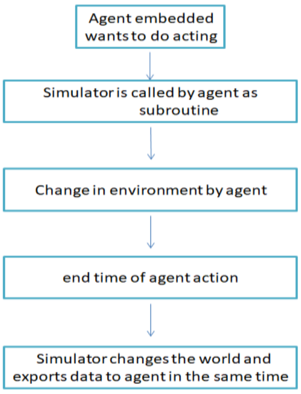

Figure 5: steps of work between agent and its environment

The responsibility of the agent designer is clear after reviewing the mechanism of action of the agent in his environment within a certain interface, the specifications of the sensing and the action that makes the designer focus on them. When the agent directly benefits from the world's information, the result is a high-fidelity model for the agent. As for processing the data (changing it) before the agent benefits from it (confusing sensing), the model is not characterized by accuracy.

Testbed	Agent/Agents	Static/Dynamism	Objects
TileWorld	Single agent	Dynamism	Tile , holes, obstacles

Table 3: characteristic of TileWorld

6.1.2 NTW (1991)

NTW [39] The 2D test bed has tiles and has no holes or obstructions. External events are caused by winds blowing on the tiles in the grid, (see Table 4). Simulators here have two characteristics:
First, the NTW test system has no implicit proportion of accomplishment that is closely resembling the thought of a score. What the agent should do and what establishes achievement is left completely to the experimenter. The second is the idea of the interface between the agent and its current world. The TILEWORLD agent calls the test system as a subroutine and gives data to and fro utilizing a common information structure(shared).The NTW agent and the world test system run no concurrently: The agent presents orders on the world, which are placed in a line(queue) and in the end executed. Administrators can be modified to fail probability : A think activity probably won't bring about the agent holding the tile, and a move may bring about the agent being dislodged to an adjoining area none than the proposed location. The agent is given no indication of whether an operator has succeeded or failed and must explicitly sense the world to ascertain the effects of its actions.

Testbed	Agent/Agents	Static/Dynamism	Objects	External events
NASA TileWorld(NTW)	Single agent	Dynamism	Tile , holes, obstacles	Winds

Table 4: Characteristics of NTW

6.1.3 MICE (1992)

MICE [41][40] is grid-situated test system, intended to help examination into organizing the problem-solving behavior of various self-ruling agents. Consists of only a number of agents, those agents can be objects like tiles, obstacles, and forest fires, (see the Table 5), the world is inhabited by only agents, diversity may be their condition. The move command is the MICE operator, from cell to neighboring cell moves the agent. To hold things by the agent, it must use the link command. The link command is a copy of the grasp operator. External events caused by agents have effects in the world (to simulate rain, a cell is wet and slippery). The fundamental difference between the mouse and its predecessors from the testbed (TileWorld and NTW) is that the MICE is of little commitment to physics in the world. A framework for testing rather than simulating this is what a MICE looks like. (Copies of Phoenix and TileWorld built by MICE designers. See[41], As instance).

Testbed	Static/Dynamism	Objects	External events
MICE	Dynamism	Agent as Tile , agent as holes, agent as obstacles	Wet cell or Slippery Cell

Table 5: Characteristics of MICE

6.2 Multi Agents in Virtual Environment

6.2.1 The PHOENIX Test Bed

PHOENIX [42] is a framework for carrying out and testing numerous self-ruling agents in a perplexing climate. The procedure is putting out fires; the world comprises of a guide with shifting territory, heights, and climate. Flames can begin at any area and spread according to the encompassing lands. Agents are putting out fires units (tractors or bulldozers as commonly) that change the lands shape to control the flames.
A distinction has to be made between agents, simulators, and the environment for Phoenix. The simulator has three core functions:
(1) to manipulation and update the map;
(2) to synchronize the action of the world and the agents, which are executed as autonomous assignments; and
(3) to accumulate information.
The PHOENIX world incorporates a portrayal of Yellowstone National Park (from Defense Mapping Agency data) and the undertakings that execute fires.
PHOENIX agents produce assignments that mimic a fire chief, a number of tractors, lookouts, helicopters, fuel big haulers, etc. Agents errands incorporate getting through the map, stopping a fire line, estimating(prediction) the course of flames, arranging the assault on the fire by a number of tractors, checking progress and distinguishing disappointments in assumptions, and recuperating from disappointment. Tasks embed themselves (by sending messages) onto a timetable kept up by the PHOENIX simulator. Agents and fires only, are types of objects in PHOENIX world, (see Table 6). However, agent and fire determine their behavior using information found in cells, each cell in the environment possessing information. In instance, the black road designed on which the bulldozers are moving quickly, and the flames escalate faster towards the designated path uphill. As for the external events, they also have an indirect effect, with the blowing of the winds, the eruption of fire accelerates.

PHOENIX contains a trial running tool that have a language for determining contents for modifying the world, fires beginning, and more varies events. It additionally takes into consideration agents conduct to be checked, delivering information documents that can be perused by information control and statistical bundles.

Testbed	Agent/Agents	Static/Dynamism	Objects	External events
PHOENIX	Agents	Dynamism	Fires	Winds

Table 6: Characteristics of PHOENIX.

6.2.2 TRUCKWORLD

TRUCKWORLD [43] is a testbed multi-agent system, it designed for reaction running theories to be tested with it, and provides supportive images (examples) for a reasoning theories about dynamism and unpredictability[44][45]. The principle responsibility is to give a reasonable world (realistic) to its clients. However, without sensors or actual impacts(physical effects). An agent is a truck comprising with 2-arms; two load coves; a number of sensors; and different parts, for example, a bunch of tires, a gas tank, and way and speed regulators. It works in an environment comprising of streets and areas. Streets associate the positions, objects populated them. The test system itself places not many limitations on the conduct of objects, it may be have degree of complex. Model objects can be showing by TRUCKWORLD , for example, Trucks can increase their fuel by using fuel barrels, they are objects, safe driving on slippery roads, trucks turn into tire chains, they are objects, and so on.
External events: Rainstorms blow frequently in the world , (see Table 7), roads become slippery and dirt roads are muddy due to these rains. Without tires to stick to muddy roads, trucks run the risk of getting into mud. Rainstorms moisten things whose action (behavior) is influenced by humidity. The events are related to a random factor and environmental characteristics (rain in hours of the day is more bearable than other hours). There were two principle objectives in designing TRUCKWORLD: (1) without focusing on a particular problem domain it gives a testbed that creates fascinating problems both in deliberative and in re-active reasoning and (2) significant limits on the sensing and effecting of agent capabilities and on the cause of building of the world. However the system allow to be extended to provide the needs of designer.

Testbed	Agent/Agents	Static/Dynamism	Objects	External events
TRUCKWORLD	Agents	Dynamism	streets and areas	Rainstorms

Table 7: Characteristics of TRUCKWORLD.

7.Discussion

Benchmarks draw us the behavior of the agent at a bright level. Communication, collaboration, lifelike, Mobility, Four typical characteristics of an agent. Intelligence is automatic for the agent because it does not exercise human intelligence, so we see the intelligent agent falls within the framework of its ability to autonomy and learn. What is the design of an agent that achieves the characteristics such as autonomy, reactiveness, proactive, social ability and so on.

A rational agent should choose an activity that is relied upon to boost its measure of performance, given the proof given by the percept succession and whatever underlying information the agent has. Environment characteristic, as fully and partial observation, continuous and discrete, and so on The classical planning paradigm accepts a climate that is both controlled and basic. Re-acting systems, for instance, manage the issue that the world can change erratically between plan time and execution time by choosing what to do at execution time as opposed to creating an arrangement preceding execution. The time consumption of planning gets significant in a world that permits impromptu changes. The Events may be happen. Loosening up this supposition makes the way toward foreseeing the plane's impact more troublesome. Pollack and Ringuette (1990) report on the TILEWORLD test bed, an unexpected dynamic system that provides controlled, observable, and traceable experiences built for adaptive agent architecture. IRMA is concerned with developing the TileWorld and in fact the TileWorld is included IRMA when distributing it to the research community.

NTW is the 2D test bed has tiles and has no holes or obstructions. External events are caused by winds blowing on the tiles in the grid. MICE is grid-situated test system, intended to help examination into organizing the problem-solving behavior of various self-ruling agents. Consists of only a number of agents, those agents can be objects like tiles, obstacles, and forest fires, the world is inhabited by only agents, diversity may be their condition. PHOENIX is a framework for carrying out and testing numerous self-ruling agents in a perplexing climate. TRUCKWORLD is a testbed multi-agent system, it designed for reaction running theories to be tested with it, and provides supportive images (examples) for a reasoning theories about dynamism and unpredictability.

8.Conclusion

The metrics that determine the agent's ability to adapt to the dynamic unpredictable environment without thinking long, these metrics are placed in an environment that simulates the real world, we must have a background on the real world specifications to build the virtual environment according to simulation controls. This environment is a testbed as TileWorld. Test beds are diverse according to the purpose for which they are designed. In other words, we have parameters in the test beds that take a picture of the agent's design and performance. From the above, it is necessary for us to have a detailed understanding of many concepts such as classical plans, various simultaneous goals, external events, time consumption, availability of resources and probability in circumstances that cannot be predicted and building models. This is what we cannot expand on in these papers, hoping that we will have a detailed pause for those research terms.

References:

[1] M. Lees, "A history of the Tileworld agent testbed," *Sch. Comput. Sci. Inf. Technol. Univ. Nottingham, Nottingham*, pp. 2001–2002, 2002.

[2] S. Hanks, M. E. Pollack, and P. R. Cohen, "Benchmarks, test beds, controlled experimentation, and the design of agent architectures," *AI Mag.*, vol. 14, no. 4, p. 17, 1993.

[3] G. J. Sussman, "A computer Model of Skill Acquisition. 133 p." Elsevier Science/North-Holland, Amsterdam, London, New-York, 1975.

[4] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Solving Large-Scale Constraint-Satisfaction and Scheduling Problems Using a Heuristic Repair Method.," in *AAAI*, 1990, vol. 90, pp. 17–24.

[5] D. S. Al-Azzawy and S. A. Diwan, "Design of Intelligent Agent Based management security system for E-government," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 9, no. 2, p. Page-131, 2017.

[6] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Auton. Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[7] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Woolley, "KAoS: Toward an industrial-strength open agent architecture," *Softw. agents*, vol. 13, pp. 375–418, 1997.

[8] C. I. Petrie, "Agent-based engineering, the web, and intelligence," *IEEE Expert*, vol. 11, no. 6, pp. 24–29, 1996.

[9] A. A. Karim and N. M. Shati, "Abnormality Detection using K-means Data Stream Clustering Algorithm in Intelligent Surveillance System," *J. AL-Qadisiyah Comput. Sci. Math.*, vol. 9, no. 1, pp. 82–98, 2017.

[10] N. R. Jennings and M. Wooldridge, "Applications of intelligent agents," in *Agent technology*, Springer, 1998, pp. 3–28.

[11] Å. Rönnbom and L. Andersson, "Intelligent Agents-A New Technology for Future Distributed Sensor Systems?," 1999.

[12] M. S. Al-Abraheemee, "Intelligent Agent for PCManagr Application," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 1, no. 2, pp. 84–94, 2009.

[13] P. Maes, "ªAgents that Reduce Work and Information Overload, º Comm." ACM, 1994.

[14] H. S. Nwana and D. T. Ndumu, "A brief introduction to software agent technology," in *Agent technology*, Springer, 1998, pp. 29–47.

[15] T. Koda and P. Maes, "Agents with faces: The effect of personification," in *Proceedings 5th IEEE International Workshop on*

Robot and Human Communication. RO-MAN'96 TSUKUBA, 1996, pp. 189–194.
[16] P. Maes, "Artificial life meets entertainment: lifelike autonomous agents," *Commun. ACM*, vol. 38, no. 11, pp. 108–114, 1995.
[17] C. G. Harrison, D. M. Chess, and A. Kershenbaum, *Mobile Agents: Are they a good idea?* IBM TJ Watson Research Center Yorktown Heights, New York, 1995.
[18] S. Franklin and A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," in *International workshop on agent theories, architectures, and languages*, 1996, pp. 21–35.
[19] H. V. D. Parunak, "Practical and industrial applications of agent-based systems," *Environ. Res. Inst. Michigan*, 1998.
[20] B. Hayes-Roth, "An architecture for adaptive intelligent systems," *Artif. Intell.*, vol. 72, no. 1–2, pp. 329–365, 1995.
[21] S. J. Russell and P. Norvig, "Artificial Intelligence-A Modern Approach, Third Int. Edition." Pearson Education, Upper Saddle River, NJ, USA, 2010.
[22] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.
[23] K. S. Løland, "Intelligent agents in computer games." Institutt for datateknikk og informasjonsvitenskap, 2008.
[24] O. A. Raheem and E. S. Alomari, "An Adaptive Intrusion Detection System by using Decision Tree," *J. AL-Qadisiyah Comput. Sci. Math. Vol*, vol. 10, no. 2, 2018.
[25] S. J. Hanks, "Projecting plans for uncertain worlds." Yale University, 1990.
[26] P. E. Agre and D. Chapman, "Pengi: An Implementation of a Theory of Activity.," in *AAAI*, 1987, vol. 87, no. 4, pp. 272–286.
[27] M. E. Bratman, D. J. Israel, and M. E. Pollack, "Plans and resource-bounded practical reasoning," *Comput. Intell.*, vol. 4, no. 3, pp. 349–355, 1988.
[28] S. J. Russell and E. Wefald, *Do the right thing: studies in limited rationality*. MIT press, 1991.
[29] T. L. Dean and M. S. Boddy, "An Analysis of Time-Dependent Planning.," in *AAAI*, 1988, vol. 88, pp. 49–54.
[30] D. S. Weld and J. De Kleer, *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann, 2013.
[31] L. Chrisman and R. G. Simmons, "Sensible Planning: Focusing Perceptual Attention.," in *AAAI*, 1991, pp. 756–761.
[32] P. Haddawy and S. Hanks, "Utility Models for Goal-Directed, Decision-Theoretic Planners," *Comput. Intell.*, vol. 14, no. 3, pp. 392–429, 1998.
[33] M. P. Wellman and J. Doyle, "Preferential Semantics for Goals.," in *AAAI*, 1991, vol. 91, pp. 698–703.
[34] L. G. Gasser and A. H. Bond, *Readings in distributed artificial intelligence*. M. Kaufmann, 1988.
[35] A. M. Law, W. David Kelton, and L. W. Koenig, "Relative width sequential confidence intervals for the mean: Relative width sequential confidence intervals," *Commun. Stat. Comput.*, vol. 10, no. 1, pp. 29–39, 1981.
[36] P. Langley and M. Drummond, "Toward an experimental science of planning," in *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*, 1990, pp. 109–114.
[37] L. A. Noor, "Predictive Model based on Logical Analysis of Data," *J. AL-Qadisiyah Comput. Sci. Math.*, vol. 6, no. 1, pp. 52–56, 2014.
[38] M. E. Pollack and M. Ringuette, "Introducing the Tileworld: Experimentally evaluating agent architectures," in *AAAI*, 1990, vol. 90, pp. p183-189.
[39] A. B. Philips and J. L. Bresina, "Nasa tileworld manual (system version 2.2)," 1991.
[40] T. Montgomery, J. Lee, D. Musliner, E. Durfee, and D. Darmouth, "Y. So. MICE users guide," Technical Report CSE-TR-64-90, Dept. of Electrical Engineering and Computer …, 1992.
[41] T. A. Montgomery and E. H. Durfee, "Using MICE to study intelligent dynamic coordination," in *[1990] Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*, 1990, pp. 438–444.
[42] D. Hart and P. Cohen, "PHOENIX: A test bed for shared planning research," 1990.
[43] D. Nguyen, S. Hanks, and C. Thomas, "The TRUCKWORLD manual," Technical Report TR 93-09-08, Dept. of Computer Science and Engineering …, 1993.
[44] S. Hanks and D. McDermott, "Modeling a dynamic and uncertain world II: action representation and plan evaluation," *J. Log. Comput.*, 1993.
[45] S. Hanks and D. McDermott, "Modeling a dynamic and uncertain world i: Symbolic and probabilistic reasoning about change," *Artif. Intell.*, vol. 66, no. 1, pp. 1–55, 1994.