Chapter 1

Introduction

In this chapter of the study, introduction is presented. Initially, the background on Artificial Intelligence (AI) and agent technologies is summarised. Then, motivation and contributions of this dissertation study are presented. Finally, the organisation of the dissertation is given.

1.1 Background

The idea of representing intelligence on machines dates back to the 18th century. Initially, one of the most popular games requiring intelligence, chess, became the first application area of Artificial Intelligence [200]. The chess game has a single objective and it has very strictly defined rules; therefore, it is rather easy to imitate on machines. In 1769, Wolfgang von Kempelen invented "Türk" which is a mechanical chess player. Although it is just a quasi automaton, "Türk" is one of the most meaningful attempts in the Artificial Intelligence field [97].

The idea of von Kempelen was so attractive that even poet and writer Edgar Allen Poe [181] wrote an article on this pseudo automaton. In the article "Maelzel's Chess Player", Poe tried to explain actions that were performed by "Türk". The first real chess playing system was designed by Torres y Quevedo. He introduced a machine that plays an end game with king against king and rook [227].

In 1920, Rossum's Universal Robots was written by Czech novelist Karel Capek. In this book, the term "robot" was coined to refer to intelligent humanoid machines. This science fiction book concerns robots that revolted against their human masters. Through this book, the term robot came to replace the terms automaton and android [66].

Even though these ideas on machines that somehow simulate intelligence were proposed before the 20th century, the term Artificial Intelligence was not coined until the 1950s. In "Computing Machinery and Intelligence", Alan Turing [224] asked if machines could think. Having been inspired by this question, Marvin Minsky, John McCarthy, Nathaniel Rochester and Claude Shannon organised the Dartmouth Conference in 1956 [229]. The proposal of the conference provided a framework for the notion of Artificial Intelligence with the following sentence [160]:

> "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

At the conference, the term Artificial Intelligence (AI) was coined by McCarthy. At the same conference, the first Artificial Intelligence program Logic Theorist was presented. This program was developed by Allan Newell and Herb Simon to expose a view on theorems in symbolic logic [81]. After these developments, AI started to become a research domain that focuses on simulating aspects of human intelligence on machines [171].

In the mid-1950s, Oliver G. Selfridge proposed the term agent which originated from the ideas of McCarthy [127]. Both researchers stated that agents are softbots that perform actions in a computer environment where they are situated. Commands are the sensors and effectors of the softbots; by using commands, they interact with the environment [240]. At that period of time, the term agent did not draw sufficient attention.

Following these developments in 1970s, researchers began to develop expert systems based on the ideas lying behind Logic Theorist. Expert systems are built in order to simulate decision-making processes of human experts. These systems are devised to assist users in complex processes which include problem analysis and solving [170]. In the subsequent decade, commercial applications of the expert systems became popular [108].

However, expert systems are just constructs that ask questions to receive answers. Based on the answers obtained, they give suggestions on decision alternatives by the help of their knowledge base. Moreover, these constructs are disembodied which means that they do not have any effect on the environment.

In 1980, a group of researchers held a workshop at the Massachusetts Institute of Technology which led to a sub-research domain Distributed Artificial Intelligence (DAI). Until that time, the AI community was focusing on intelligence as a whole. However, DAI asserted that intelligence is constituted of distributed properties like learning, reasoning, and so on. After this workshop, the term agent attracted the focus of attention. This development gave rise to agent technologies which became a commonly approved research area [119].

Even though the term agent still refers to situated entities which interact in a computer environment, the meaning of it has changed slightly. Today, the term implies entities that represent some aspects of intelligence. In spite of the fact that there is a debate on what the term agent means, Jennings et. al. [119] put forward a commonly acceptable definition as follows:

> "A computer system, situated in some environment that is capable of flexible autonomous action in order to meet its design objectives."

In this definition, Jennings et. al. [119] put forward three attributes that an agent should have: situatedness, flexibility and autonomy. They state that the term situatedness implies entities that are capable of obtaining sensory data and performing actions to change the environment where they are embodied.

They explained the term flexibility as a capability for performing flexible actions. They further elaborated this attribute by asserting its three constituents: responsive, pro-active and social. Entities that can understand their environment and respond to the changes that occur in their environment are called responsive. Pro-active entities can perform actions and take initiatives to achieve their objectives. Finally, the term social implies systems that are able to interact with other entities and also help others in their activities.

By the term autonomy they were referring to those entities that can perform actions without the assistance of other entities. Moreover, those entities have the ability to control their internal state and actions. Russell and Norvig [194] gave a stronger sense of autonomy by adding that those entities should also have the ability to learn from experiences. To achieve stronger autonomy, Luck and D'Inverno [150] suggested that the agents should have motives to allow them to generate goals.

In addition to strong autonomy, to develop human-like intelligent agents, the research has indicated the importance of emotions [22]. Many researchers have stressed that the core requirement for believable agents is a capability for affect display. Therefore, to achieve human-like intelligence, intelligent agents should also be capable of affect display by employing an emotion model.

Some researchers stated that agents could not be limited to these four core attributes; they added that agents can also have some additional properties based on their design objectives. Bradshaw [34] listed these properties as reactivity, knowledge level, inferential capability, mobility, temporal continuity, and personality.

It must be noted that while developing agents, the commonly approved fundamental attributes that a believable agent should have are situatedness, autonomy, flexibility, and affect display. Assigning other attributes to agents depends on their design objectives.

Today, the notion of an agent has become central to Artificial Intelligence. Therefore, agent technologies hold promise to recreate intelligent behaviour in all respects while trying to imitate these core attributes [239].

Furthermore, agent-based solutions evolved in such a way that they started to become evolutionary societies. Evolutionary agent societies expand agent technologies further [190]. In such agent societies, agents are capable of seeking each other and cooperating in a parallel and distributed manner. Channon and Damper [1] state that evolutionary emergence is a key to generating social agents. In order to achieve this aim, they suggest adopting neural networks; since, neural networks suitable for intelligent behaviour. Along with the core attributes of intelligent agents, evolutionary agent societies provide a promising way to imitate social phenomenon in the computer environment.

Recently, artificial neural networks have been adopted in robotics and evolutionary algorithms. Evolutionary robotics, based on evolutionary agents, are being used as a scientific tool for studying models of cognition. Such systems can help re-orginise the conception of human-intelligence [109].

While conceptualising agent architectures, most of the researchers utilize the intentional notion which provides a theoretical infrastructure for agency. They use this notion while they are trying to satisfy core attributes of intelligent agents. While explaining intentional stance, Dennett [71] states three levels of abstraction which help explain and predict the behaviours of entities and objects: (1) physical stance; (2) design stance: and (3) intentional stance.

Dennett [71] puts forward physical stance to explain behaviours by utilizing concepts from physics and chemistry. In this level of abstraction behaviours of entities are predicted by considering the knowledge related with things like energy, velocity and so on. As an example, it can easily be predicted a ball would fall down to the ground if it is released from the top of a building. It is due to the fact that there is gravity on earth and the behaviour of the ball can be anticipated by considering this physical principle.

Design stance is a more abstract level and at this level of abstraction, the manner of acting is explained and predicted through biology and engineering. At this level, things such as purpose, function and existential attributes are taken into account. It can be predicted that the speed of a car is going to increase whenever we press on the gas pedal. It is because of the fact that it is known that the gas pedal is made for increasing the speed of the vehicle. In this case, design stance is utilized to understand the behaviour of the car.

Last stance is intentional stance which is the most abstract level. At this level of abstraction, an attempt is made to understand and predict the behaviours of software and minds by considering mental concepts such as intention and belief. As an example, if a bird flies away while a cat tries to catch it, it can be understood that the bird desires to live. This can be comprehended by taking intentional stance into account.

As stated formerly, the theory of agency is based on the intentional stance. According to this, entities are treated as rational agents. Behaviours of agents are predicted by considering what beliefs an agent ought to have based on its purpose in a given condition. Afterwards, an attempt is made to predict what desires the agent ought to have, based on the same conditions. Finally, it is predicted that a rational agent will act to achieve its goals under the guidance of its beliefs. From this point of view, practical reasoning helps towards an understanding of what the agent ought to do based on the chosen set of beliefs and desires.

1.2 Motivation and Aims

In the literature, there are several architectures that attempt to mimic human-like intelligence. These architectures fail to satisfy the aforementioned core attributes all at once. Most of those studies utilize the intentional notion to simulate human intelligence. To achieve strong autonomy, some of those studies employ learning approaches while others adopt motivation theories. To simulate affect display, a few studies employ an emotion model. In this respect, there is no general approach covering all aspects of the problem. To achieve this aim an approach should employ a learning model, adopt motivation theories, and utilize an emotion model. Therefore, the first research question addressed in this study is: "What kind of an agent architecture is required to cover the core attributes of agents to develop a believable agent?". The first major objective of this study is to seek an answer to this research question.

Besides, the existing approaches mainly adopt Means-Ends Analysis or some probabilistic techniques. If fixed values are used to compare different plan alternatives, this results in choosing the same plan alternative under the same conditions. This is not sufficient to explain intelligent behaviour. On the other hand, action flexibility provided by adopting these techniques is similarly not sufficient. It is because of the fact that for a given set of probabilities the produced behaviour is quite predictable. Therefore, these approaches are inadequate in simulating the unpredictability of human behaviour even if the probability matching techniques are utilized.

The question of "How does an agent architecture simulate unpredictability of human behaviour?" still remains unanswered. The second major objective of this study is to find an answer to this research question.

In the light of this discussion, this study proposes the development of an agent architecture that satisfies the aforementioned core attributes. The other objective of this study is to simulate the unpredictability of human behaviour. With this respect, the major aim of this study is twofold: (1) to develop a general framework that covers the core attributes of intelligent beings in order to support the development of believable agents; and (2) to develop a more realistic decision-making mechanism while simulating the unpredictability of human behaviour. The major aim in addressing these issues is to achieve a general approach to simulate human-like intelligent behaviour. By addressing these issues, this thesis study attempts to develop a general agent architecture to support the development of believable agents with a highly realistic decision-making mechanism.

1.3 Contributions

In this study, a new approach is proposed to simulate the intelligence of intelligent beings. This approach is based on three fundamental assumptions. Firstly, it is assumed that all intelligent entities are intentional. Secondly, all of these beings are driven by some motives. Thirdly, intelligent behaviour is produced in accordance with causality.

For this purpose, the intentional notion and theories of needs are combined. It is proposed that motives, particularly the needs of intelligent entities, drive them to satisfy their motives while acting intentionally. In the proposed study, needs (i.e., motives) are introduced as a nexus. In this manner, the satisfaction obtained by meeting needs provides the means to measure different alternatives and select among alternatives. The satisfaction degrees are normally distributed random numbers lying between 0 and 1 with certain mean values. By providing a degree of randomness in the decision-making process, the proposed approach promises to simulate the unpredictability of intelligent behaviour. In this manner, the new approach provides enhanced action flexibility.

In this approach while explaining motives, theories of needs are adopted. It is proposed that decision-making is a process in which intelligent entities produce effects which yield intelligent behaviour due to some causes. These causes are the conditions which result from the observation of either the internal state or the external world. In the decision-making process, the nexus is offered as needs which provide a metric to measure different alternatives.

Along with these issues, it is proposed that emotions emerge when the needs of intelligent beings are satisfied or not satisfied. According to this viewpoint, while satisfaction of lower level needs triggers primitive emotions, meeting higher level needs results in more complex emotions. Therefore, every need in the hierarchy is associated with two different emotions. While one of these emotions is positive, the other one is negative. Whenever a particular need is sufficiently satisfied, it results in the generation of a positive emotion. If a particular need is not satisfied sufficiently, it results in the generation of a negative emotion.

In this approach, the effect of emotions on intelligent behaviour is similarly illustrated. To illustrate this effect, it is proposed that emotions have a direct influence over the order of needs. If a particular need is sufficiently satisfied, it results in a positive emotion. In addition, if a particular need is more than sufficiently satisfied then it is going to result in strong emotions. Therefore, emotions are categorised as regular and strong emotions. Strong emotions change the order of the associated need while regular emotions do not. From this point of view, a strong positive emotion can bring the associated need to a lower level. While a strong negative emotion can force the associated need to move to a higher level.

Under the light of these ideas, it is proposed that the term intelligence refers to an abstract notion to express cognitive processes of autonomous, embodied, flexible, and social entities which can display affect and learn while they perform activities intentionally that are motivated by their needs. The new definition is meant to explain behaviours of all intelligent entities which are human-beings, animals and agents.

Based on this framework, a new agent architecture called Reactive-Causal Architecture (ReCau) is proposed. This architecture is a general purpose architecture which can be adopted to mimic human intelligence. This architecture is not meant to be an efficient one but rather it is meant to illustrate the proposed approach.

The major contribution of the current study is that it provides a general approach that covers the core attributes of believable agents. The other contribution of this dissertation is that it proposes a new agent architecture ReCau which provides a guideline to implement the proposed approach. This research study also contributes to development of highly realistic decision-making mechanism to mimic unpredictability of human-intelligence.

1.4 The Organisation of the Thesis

The organisation of this dissertation is organised as follows. In the second chapter, the background on intelligent agent technologies is given. The theoretical infrastructure of intelligent agents related to intentional notion is presented first. Then, issues related to multi-agent systems are presented. To support the flexibility attribute, agents must be capable of communicating and cooperating with one another. Therefore, in this chapter agent communication and agent cooperation approaches are reviewed. Finally, a review on agent oriented programming languages is provided.

In the third chapter of the study, the literature on agent architectures is discussed. Currently, agent architectures are mainly categorised in four groups: (1) deliberative, (2) reactive, (3) hybrid, and (4) cognitive. In accordance with these categories existing architectures are explored in this chapter.

In the fourth chapter, the details of the proposed approach are presented. As previously stated, in the proposed approach, an attempt is made to combine the intentional notion and the theories of needs. In this respect, while explaining the foundations of the proposed approach, the background on the theories of motivation is also presented.

In the fifth chapter, the details of the proposed architecture called Reactive-Causal Architecture (ReCau) are elaborated on. ReCau is a three-tiered architecture consisting of reactive, deliberative and causal layers. This architecture is a general purpose one which can be used to simulate intelligent human behaviour. In this chapter, the general structure of the architecture is explained. Then the mechanisms and components employed in the architecture are elaborated on. Finally, ReCau is compared with a number of existing architectures.

In the sixth chapter of the study, two simulation studies are presented. The first simulation is performed to illustrate the action flexibility provided by ReCau. The second one called the radar task simulation is an organisational decision-making simulation. To illustrate the decision-making mechanism of the architecture, the radar task simulation is undertaken. In this task, a collection of agents attempts to determine whether a blip on a radar screen is a hostile plane, a civilian plane, or a flock of geese. This simulation was also performed by a few existing architectures; therefore, it allows for comparison of ReCau with those architectures.

In the last chapter of the study, discussions and conclusions are given. In addition, the future research directions are discussed.

Chapter 2

Intelligent Agent Technologies

In this chapter of the study, the literature related to intelligent agent technologies is reviewed. The background on intentional notion which forms the theoretical infrastructure of agency is explored first. Then, agent communication and agent cooperation approaches are presented. Finally, programming languages that are used to develop intelligent agents are summarised.

2.1 Agent Theories

There are a few theories on formally representing the properties of the agents [240]. The most common theory for representing the properties of agents is to utilize the intentional notion [239]. According to this view, agents are designed as intentional systems by ascribing some mental qualities like beliefs, desires, likes and dislikes to machines.

The notion of intentionality was first introduced by Brentano [37]. While putting forward this term, Brentano aimed to form a criterion in order to distinguish mental from physical phenomena. In his book, he did not try to develop systematic accounts of intentionality.

Inspired by the ideas of Brentano, intentional stance was coined by Dennett [69]. He put forward three levels of abstraction in order to understand and predict the behaviours of systems like animals and computers. These three levels are called physical, design and intentional stance.

For instance, the actions of a chess playing computer can be predicted by using these three abstraction levels. The first level of abstraction is called physical stance. By utilizing this stance, predictions are based on the physical state of the objects. The predictions are made by applying knowledge related to the laws of nature. From this stance, it is not possible to understand the actions of a chess playing computer. However, malfunction of a computer system can be understood at this level of abstraction.

Dennett [69] states that in principle, the actions taken by a chess playing computer can be predicted from physical stance. However, he says that it is pointless to attempt the prediction of the behaviours of such systems by employing this stance. Moreover, it is suggested to reserve physical stance for instances of system breakdowns.

Secondly, there is the design stance which takes elements such as purpose, function and existential attributes into account. If one knows the design details of a computer system and the programs that are installed on it then the responses of the system can be predicted by following computation instructions. The predictions are true provided that the computer performs as designed, which implies that there should be no breakdowns.

These predictions are actually based on the notion of function. Dennett [69] suggests adopting the design stance to try to understand the responses of mechanical objects. At this level of abstraction, the predictions are made solely based on the knowledge and/or assumptions on the functional design of the system. Such predictions are irrespective of the physical constitution or condition of a particular object.

Dennett [69] suggests intentional stance to predict the behaviours of a chess playing computer instead of the design stance or the physical stance. He states that these systems become so complex that their behaviours cannot be predicted from the design stance or the physical stance.

If intentional stance is followed, given the rules and goals of chess, the responses of a computer can be predicted by finding its best or most rational move. Such predictions are based on two basic assumptions:

- The computer functions as designed, and
- Design is optimal and the computer will choose the most rational move.

In this manner, a computer can be seen to be rather like an intelligent being and its movements are predicted. By assuming that a chess playing computer is an intentional system, its actions can be understood; therefore, while developing such systems it is reasonable to develop them as an intentional system which has beliefs, desires, intentions, and goals.

Dennett [70] suggests that intentional systems have different levels of intentionality. First-order intentional systems have only beliefs and desires. Second-order intentional systems have beliefs and desires on their own beliefs and desires and those of others. By this assertion, he does not only infer beliefs and desires but also means other intentional stances such as intentions and obligations.

McCarthy [157] goes further by stating the conditions under which intentional stance can to be ascribed to machines. According to his opinion, it is more appropriate to use intentional stance for agents and computers when such an ascription expresses the same information on an intelligent being.

Within the frame of these references, the behaviours of intelligent entities are predicted and explained through the attribution of attitudes such as believing, desiring, hoping, fearing and so on. According to intentional stance, the attitudes, that form intentions, consist of information attitudes and pro-attitudes. While information attitudes cover knowledge and belief; pro-attitudes include desire, choice, commitment, intention, obligation and so on [239].

Information attitudes are related to the information on a situated environment. Information attitudes can be categorized in two groups: knowledge and belief. Knowledge can be explained as true information. Belief is something believed or accepted as true. Beliefs are the information that is obtained by sensing, learning, etc... Pro-attitudes which include desire, intention, obligation, commitment, and choice guide actions [240].

Moreover, Dennett [70] stresses that intentional notion explains the human intelligence somehow. However, from the functionalist point of view it lacks in clarifying all aspects of the human intelligence. In particular, the emergence of intelligent behaviour cannot be explained by only intentional notion. But it must be emphasised that intentional stance provides a very good theoretical infrastructure for agency.

Based on this infrastructure, agent theories provide relations in between several attributes while trying to explain following issues:

- Relations between information and pro-attitudes,
- Change in the cognitive state of an agent over time,
- Interaction between cognitive state of an agent and the environment in which the agent is situated, and
- Performing actions in guide of the attitudes.

Moore [161], one of the pioneer agent theorists, attempted to develop theory of knowledge and action. He researched on the things that should be known by an agent. He studied on determining pro-attitudes regarding actions. By his model, he recommended which actions should be performed in case of incomplete information. In this manner, he determined how agents should achieve their objectives with incomplete information.

At the same period of time, Cohen and Levesque [59] developed the theory of intention. While developing their theory they were inspired from Bratman [35]. According to intention theory, following criteria are proposed by Cohen and Levesque:

- Intentions cause problems that are needed to be solved by agents,
- Intention filtering is required in order to prevent conflicts,
- Agents perform actions until they reach their goals even if they fail in some cases,
- Agents believe that their intentions are possible, and
- Agents believe that they bring about their intentions.

According to these criteria, Cohen and Levesque proposed a new approach. They developed a logic of rational agency and partial theory of rational action. Logic of rational agency is defined in terms of relations between other modal logic operators.

There are a lot of proposals on determining combination of attitudes that are required to build a rational agent [239]. The most popular of these approaches is called Belief, Desire and Intention (BDI) which is put forward by Rao and Georgeff [185]. As it can be understood from its name, this approach includes three components: Belief, Desire and Intention.

Beliefs correspond to the information that an agent has about itself and its environment. Beliefs can also include inference rules, allowing an agent to generate further beliefs. Typically, when implementing the BDI approach, beliefs can be stored in a database (belief base). Here the term belief implies that the relevant information may or may not be true and beliefs can change in time.

Desires represent the possible alternatives that can be chosen by an agent. In other words, desires represent objectives or situations that an agent would like to accomplish or bring about. Instead of desires, in some cases the term goal can be used. However, the term goal adds further restrictions that the set of goals must be consistent.

Furthermore, the attitudes are defined in terms of beliefs and desires. In the BDI approach, intentions are the choices of an agent. Intentions are the desires to which an agent has committed. In an implemented system, this means that the agent has begun executing a plan. Plans are sequences of actions that an agent can perform to achieve one or more of its intentions. In some cases, plans may include other plans.

In the BDI approach, practical reasoning occurs by updating beliefs continuously and comparing possible alternatives; therefore, alternatives are filtered to determine new intentions. According to these intentions, plans are made and the actions are performed in accordance with the plans.

Another approach is proposed by Singh [204, 205, 206, 207]. He developed logical infrastructure for representing intentions, beliefs, knowledge, know-how and communication. His model of intentions and beliefs is based on Discourse Representation Theory (DRT) [124]. In the traditional natural language semantics, only individual sentences are examined. But the context of a dialogue also plays a critical role in the meaning. Therefore, DRT is put forward to represent language for examination of contextually dependent meaning in discourse.

Recently, Wooldridge [237] extended the logical framework of the BDI approach to define the Logic of Rational Agency (LORA). LORA allows representing and reasoning on beliefs, desires, intentions, and actions of agents. By this study, Wooldridge explained how beliefs, desires, intentions, and actions change over time. Moreover, he presented two different perspectives of the LORA for an individual agent and multiple agents.

2.2 Agent Cooperation Approaches

Agent theories are related to isolated components of agents. However, intelligent-beings are not isolated from others. They interact, communicate and cooperate with each other. In this subsection, the agent cooperation approaches are explored.

Multi Agent Systems (MAS) focus on building systems that include autonomous entities. The researches within the MAS domain are interested in the behaviours of the autonomous agents that aim to solve particular problems. In multi agent systems, each agent has incomplete knowledge and ability to solve a problem. In these systems, there is no global control over the system. Moreover, the data is distributed and computation is asynchronous. Multi agent systems provides interaction between systems, manages and controls distributed knowledge [119].

Multi agent system researchers mainly focus on solving the following issues [119]:

- Designing a system that includes many agents and assigning problems to those agents,
- Formulating interaction and communication between agents,
- Defining relations between local and global decisions,
- Achieving coordinating among autonomous agents,
- Solving intention conflicts among agents, and
- Improving efficiency in local computation.

When planning a single agent the objectives, the abilities and the environmental constraints are to be evaluated. However, when designing a multi agent system, the constraints regarding to each agent, the decisions that are given by a single agent and their effects on the other agents and predicting the undetermined environment become the key issues. Therefore, the main aim is to enable the agents to cooperate to achieve a common goal. Cooperative behaviour allows the agents to promote coherent system behaviour [221].

Among the studies of distributed artificial intelligence, one of the pioneering studies was on a group of agents that focus on common objectives. The agent interactions are directed by the cooperation strategies that assist in developing the performance of all agents. The pioneering studies on distributed planning used the "complete planning before action approach". According to this approach, in order to develop a coherent plan, the agents must be aware of sub-goal interactions and either avoid them or else resolve them [119].

Georgeff [95] is one of the researchers who adopted this approach. He proposed a method for synthesizing the plans of multiple agents into single agent plans. In this manner, the agents can synchronize their activities and avoid the conflicting interactions.

Lesser [144] proposed another approach for task decomposition and agent interaction called Functionally Accurate Model (FA/C). This model is meant to resolve sub-problem interdependencies. In this model, the agents do not need to have all the information to solve sub-problems, and the agents interact through an asynchronous, co-routine exchange of partial results. The FA/C enables an agent to behave in an uncoordinated manner.

The FA/C model paved the way to develop distributed control schemes for agent coordination. Partial Global Planning is an approach to coordinate the agents dynamically. In this approach, the agents communicate their plans and goals. Through these communications, the agents learn the intentions of each other. In this model, the agents are cooperative; therefore, they adjust their plans [119].

In the area of the agent cooperation, another approach is modelling teamwork explicitly. This approach is particularly helpful in dynamic environments. In such environments the agents may fail or find new opportunities. In these types of situations, the team should be able to observe its performance and reorganise accordingly [119].

Another approach called joint intentions framework focuses on representing the team's mental state called a joint intention. If the team members commit to complete an action together, the team is in the intention of cooperatively completing the action. The commitment protocol synchronizes the beliefs and the commitments of the team to complete a team task [119].

The interaction between the agents is called negotiation in the domain of self interested Multi-Agent Systems. Negotiation is proposed as a means for agents to communicate and cooperate. Negotiation refers to a method that involves communications to solve plan changes, to assign tasks and to overcome the constraint violations centrally. These conflicts should be resolved by self interested agents in such circumstances that there are incomplete information and bounded rationality. Besides, the agents should communicate and exchange their proposals and counter proposals [119].

One of the most important issues is reaching agreements among self-interested agents. Negotiation and argumentation are the most important capabilities that an agent should have to reach an agreement. Negotiation scenarios require particular mechanisms (i.e. protocol) which are the rules of encounter between agents [238].

In a multi agent system, the agents need to cooperate on given tasks. In some cases, the agents need to share tasks. Contract Net Protocol is the first task sharing protocol that is used for task allocation. In this approach, each agent can be either a manager or a contractor of a particular task(s). Whenever an agent gets a composite task or cannot solve its task, the agent breaks the problem down into sub-tasks. Then it advertises the sub-task(s) to the contract net as a manager. Potential contractors send their bids to the manager. Then the winning contractor(s) are given the task(s) [213].

One of the pioneering studies on self interested agents is Persuader. The Persuader system provides mechanisms to modify the plans, the goal and the behaviour of the other agents. In this manner, this system increases cooperation among agents to find a global solution. This system operates in the labour negotiations domain [221].

The Persuader system which is inspired by human negotiation involves a union agent, a company agent and a mediator agent. The negotiation is on issues like wages, pensions, seniority, and so on. The Persuader models the iterative exchange of proposals and counter-proposals in order to reach an agreement between the agents. Each agent has a multi-dimensional utility model which is private and different from the others [119].

Another pioneering study based on game theory is performed by Rosenschein [191, 192]. Game theory is a way of analysing the decision-making process when there is more than one decision-maker. Each agent's payoff depends on the actions taken by other agents. The actions of an agent depend on its beliefs on what the others do. What the other agents do depends on their beliefs on what each agent does. Based on these beliefs, in a game each agent tries to find the optimum outcome for themselves [186].

In the Rosenschein's study, the agents reason about the alternatives in order to find the alternative that has the maximum payoff. After finding the alternative with the maximum payoff, the agents select the alternative with the maximum payoff [119].

Kraus et. al. [132] examined the problems of resource allocation and task distribution among multiple autonomous agents. They proposed a negotiation model which takes time spent on the process into account. In this manner, they aimed to achieve efficient agreements without delays even if there are changes in the environment.

Another approach for creating cooperation to solve problems is result sharing. In result sharing, the agents cooperatively exchange information whenever a solution is developed. This process proceeds while the problem is being solved. Typically, the results progress from smaller to larger, more abstract solutions. [238].

Harandi and Rendon [106] distinguished three different basic modes of organisations as master-slave, the society of peers and the federation of autonomous agents. These organizational modes are used to create a cooperation structure in a multi agent system. In their study, they also stated that there are some derived organizational modes out of these three basic modes.

The master-slave cooperation concept is another approach in multi agent systems. In these systems, there are master and slave agents. The role of the master agent is to create cooperation among the other agents by having full responsibility for the goal and controlling the resources. The other agents are fully committed to the goals assigned to them and they can only use the resources allocated to them. All interactions are resolved by the master. Therefore, the others do not interact with each other.

Another approach is called the society of peers. This is a democratic group in which privileges and constraints are distributed. All decisions are made through negotiations. The responsibility for achieving a goal is shared among the agents that are assigned to that goal.

Another approach for cooperation is the federation of autonomous agents. In such systems, the agents are loosely coupled and they are somewhat independent. However, they still have central control at certain degree to achieve a final solution. Autonomy is granted to the agents; since, they can plan their goals, take the course of action they choose to achieve the goal. In addition, in this approach, the agents determine their level of interaction with the others.

Changhong et. al. [54] stated that there are two cooperation structures: the complete cooperation structure and the incomplete structure. A cooperation structure is said to be complete cooperation structure for agent i and goal g if and only if either [75]:

- "agent i has been delegated the goal g, and i is capable of g"; or else
- "agent i has delegated each immediate sub-goal g_0 of $g^{'}$ of g to some agent j , and the cooperation structure $(C; l)$ is complete for agent j and goal g_0".

d'Inverno et. al. [75] stated that not all the structures are cooperation structures. They indicated that a structure is a cooperation structure only if and only if:

- There are at least two cooperating agents,
- Each agent is connected to another through a cooperation over goals between them, and
- The agents do not delegate goals to the others.

Recently, Sioutis and Tweedale [208] studied on agent cooperation and collaboration. They stated that the existing implementations of the Multi Agent Systems define highly structured teams. They explained that a team is created by defining and assigning roles of the members (i.e. the agents in the system). They underlined the fact that in order to effectively form the agent teams, communication, negotiation and trust are the key issues. Therefore, research on cooperation focus on these issues. When creating teams, the basic modes and derived modes of organisations are adopted.

2.3 Agent Communication Approaches

To create cooperation among agents the communication becomes the key factor. Blackboard systems can be considered as the pioneering approach to agent communication. Blackboard-based problem solving is usually explained by the following example [63]:

> "Imagine a group of human specialists seated next to a large blackboard. The specialists are working cooperatively to solve a problem, using the blackboard as the workplace for developing the solution.
>
> Problem solving begins when the problem and initial data are written onto the blackboard. The specialists watch the blackboard, looking for an opportunity to apply their expertise to the developing solution. When a specialist finds sufficient information to make a contribution, she records the contribution on the blackboard, hopefully enabling other specialists to apply their expertise. This process of adding contributions to the blackboard continues until the problem has been solved."

The basic components of the blackboard systems are knowledge sources, a shared blackboard and a control component. The knowledge sources contain the knowledge required to solve the given problems. The blackboard is a global database which contains input data and partial solutions, alternatives, final solutions, and control information. The control component decides on the course of problem solving and managing resources. With this structure, blackboard systems support agent communication for problem solving.

The first blackboard system was called the Hearsay speech understanding system [187]. One of the researchers in the Hearsay project was Lesser. He and Fennel studied on exploiting parallelism in blackboard systems [83]. Their research exposed a bottleneck in the classical blackboard model which implied that at a given time only one thread can be written on a blackboard. By enabling multiple blackboards Lesser and Erman overcame this problem. They allowed multiple blackboard systems to communicate by message passing [145].

In early object oriented programming systems, communicating by message passing was the key idea [238]. Today message passing can be used as another approach for the agent communication. By using the message passing approach agents can send and receive messages. The messages can contain bytes, complex data structures or segments of codes.

The pioneering theory for the communication between the agents is the speech act theory that is put forward by Austin [14]. He proposed some performative verbs like request, inform, and promise for corresponding to different types of speech acts. He indicated three different aspects of speech acts [238]:

- The Locutionary Act: The act of making an utterance (For instance, saying "Please prepare a cup of coffee" is the locutionary act),
- The Illocutionary Act: The action is performed while saying something (For instance, saying "He requested me to prepare a cup of coffee" is the illocutionary act), and
- The Perlocution: The effect of the act (For instance, saying "He got me to prepare a cup of coffee" is the perlocution).

For successfully completing performatives the required conditions are called felicity conditions [238]:

- "There must be an accepted conventional procedure for the performative, and the circumstances and persons must be as specified in the procedure",
- "The procedure must be executed correctly and completely", and
- "The act must be sincere, and any uptake required must be completed, insofar as is possible."

A few years later, the speech act theory is further improved by Searle [198]. The basic axiom of the theory is communicative expressions which are acts that resemble the physical acts. The speech acts are performed by a speaker to cause a desired change with her intention which the speaker brings about in the world. In this manner, the act of the speaker tends to cause change in the mental state of the listener.

Searle systematically classified the types of the speech acts as follows [238]:

- "Representatives: A representative act commits the speaker to the truth of an expressed proposition."
- "Directives: A directive is an attempt on the part of the speaker to get the hearer to do something."
- "Commissives: Commit the speaker to a course of action."
- "Expressives: Express some psychological state."
- "Declarations: Effect some changes in an institutional state of affairs."

Later, Cohen and Perrault [61] attempted to develop a speech act theory. They proposed to model speech acts in a planning system like operators. In this manner, speech acts are treated similarly to the physical actions. This approach is called the plan-based theory of speech acts.

In 1990, Cohen and Levesque [60], developed a theory to model speech acts as actions performed by rational agents. This model of rational action was built on their theory of intention [59]. Today, most of the Agent Communication Languages (ACL) are based on these speech act theories.

Even though, it is not directly based on the speech act theory, the most commonly known agent communication language research is Knowledge Sharing Effort funded by the Defence Advanced Research Projects Agency (DARPA). With this research two languages are developed: Knowledge Interchange Format (KIF) and Knowledge Query and Manipulation Language (KQML) [240].

Knowledge Query and Manipulation Language is an agent communication language to standardise the communication among agents. KQML defines performatives for knowledge retrieval, insertion or deletion. The communication facilitators of KQML coordinate interactions of the agents. In this manner, KQML supports knowledge sharing [103]. However, KQML does not provide the means to deal with the content part of messages [238]. In the 1990s, several different versions of KQML were proposed. All of these versions include different collections of performatives.

Naturally, all agents have a different internal representation of knowledge. Therefore, the agents attribute knowledge to other agents. The term virtual knowledge base implies the attributed knowledge of other agents [238].

KQML have many different implementations which cannot interoperate. The semantics of KQML was not thoroughly defined. The set of KQML performatives are too large. Based on such criticisms, it became inevitable to undertake new research on agent communication languages.

Due to the criticisms towards KQML, the Foundation for Intelligent Physical Agents (FIPA) proposed an agent communication language which is called FIPA-ACL. The syntax of FIPA-ACL is very similar to the KQML syntax. FIPA-ACL is based on the speech act theory and it contains two distinct parts: a communicate act and the content of the message [104]. Unlike KQML, FIPA-ACL has very well defined semantics. JADE is one of the agent-oriented development platforms that implements FIPA-ACL.

The Knowledge Interchange Format is a content language designed to interchange the knowledge among disparate computer systems. It is capable of representing first order predicate logic. The syntax of KIF is based on common LISP [94]. KIF is proposed to form the content parts of KQML messages.

Standard Upper Ontology Knowledge Interchange Format (SUO-KIF) is developed to support the upper merged ontology by simplifying KIF. Like KIF, SUO-KIF is a language to author and interchange the knowledge between different entities [177].

There are some other content languages like Semantic Language and Extensible Mark-up Language (XML). These content languages can also be used as agent communication languages for the content which is embedded in the messages of the agents.

As a conclusion, in order to provide flexibility to an agent, it must be able to communicate and cooperate with other agents. To create cooperation among agents the communication becomes the key factor. Even though there are number of communication approaches, still there is no standard communication language. In addition, the simple message passing gives higher flexibility to the researchers while implementing their architectures.

2.4 Agent Oriented Programming Languages

Based on the theories and by following the agent architectures many applications have been developed. To develop these applications, it is expected to use a variety of software tools. To develop intelligent agents, concurrent object languages are originally utilized. Some instances of concurrent object languages are concurrent METATEM, and TELESCRIPT, Placa, April, ConGolog, May and Able [240].

Concurrent object languages that are the ancestors of agent languages are developed to execute objects running concurrently and autonomously. These systems can send messages to other objects with some internal state which is indirectly accessible to the environment. The Actor model and Actor-Based Concurrent Language (ABCL) system are the first instances of concurrent object languages. Without needing others, the actors form the autonomous components of interacting computing systems that communicate by asynchronous message transfer [240].

Many applications of the Belief, Desire and Intention approach are developed. As explained in the review, such systems are the instance of practical reasoning. They gave a basis for one of the pioneering agent oriented programming language that is proposed by Shoham [203]. Shoham's agent oriented programming language focuses on the social viewpoint of computation.

Agent oriented programming is directly based on programming agents in terms of intentional notions. Shoham [203] proposed that three components are needed to develop a complete agent oriented programming system. The first component is recommended as a logical system that defines the mental state of an agent. To program agents, the second component is defined as an interpreted programming language. The last component is called the agentification process by Shoham to imply compiling agent programs. The first developed agent oriented programming language is called AGENT0 system. Belief, commitment and ability were the three modalities that are covered by this system. This system is only intended as a prototype.

The first commercial agent language is developed by General Magic Inc. and it is called Telescript. This technology covers many methods and notions. It provides a language based environment to develop agents. The places and the agents are the two key concepts of this system. The agents that can be developed by this system are applications of customers and providers in an electronic marketplace [240].

One of the pioneering agent-oriented programming languages is AgentSpeak(L). This is one of the best known languages based on the BDI architecture. AgentSpeak(L) is an abstract logic-based language that allows agent programs to be written and interpreted. This language is proposed by Rao [184].

In addition to these pioneer programming languages, today there are mainly four agent-oriented programming languages based on the Java platform. The most commonly known of these programming languages is called Java Agent DEvelopment Framework (JADE). JADE is mostly used for developing multi-agent systems. It enables the programmer to develop applications based on the peer-to-peer intelligent agent approach and solve complex problems in a distributed way. JADE is in compliance with the Foundation for Intelligent Physical Agents specifications for interoperable intelligent multi-agent systems. The FIPA is an organisation to develop standards for generic agent technologies [26].

Another agent-oriented programming language is JACK [115]. JACK uses component based approach and it extends Java with agent oriented concepts. It enables users to develop agent knowledge bases and databases. It incorporates graphical design tools and plan reasoning can be laid out using simple diagrams. Besides, these plans can be traced graphically at run time. It also allows non-programmers to outline the reasoning process in natural language.

Another Java based agent-oriented programming language is JASON. It is the extended version of the AgentSpeak(L) and provides a platform for the development of multi-agent systems. Moreover, it enables the implementation of reactive planning systems according to the BDI architecture. It provides a library of essential internal actions. It also provides a speech-act based inter-agent communication mechanism [30].

An Abstract Agent Programming Language (3APL) is a programming language for cognitive agents. It provides programming constructs for implementing beliefs, goals and plans of an agent. It is capable of performing belief updates and communication actions. It provides a set of practical reasoning rules to update or revise the goals of agents [68].

There are also some agent development toolkits like ZEUS, FIPA-OS, TRYLLIAN and SimAgent [90, 209]. ZEUS provides a set of software components and tools to develop an agent system. It enables programmes to rapidly develop multi-agent applications. ZEUS has general purpose planning and scheduling mechanisms to easily develop agents.

FIPA-OS is another toolkit which enables the rapid development of FIPA compliant agents. FIPA-OS is being used in a number of European Collaborative projects. FIPA-OS is being improved as an open source project.

Another noteworthy agent development toolkit is TRYLLIAN. It enables task oriented programming. The communication between components can be performed with the hypertext transfer protocol, secure hypertext transfer protocol, or Java message service. It provides integration with the Java platform application servers and with web services.

SimAgent toolkit is developed as part of the Cognition and Affect project. It enables programmes to rapidly develop prototypes. It also supports object-oriented techniques. It enables the development of interacting agents in environments of various degrees and kinds of complexity [209].

In addition to these toolkits, there are agent simulation software tools such as Agent-Object-Relationship Simulation (AOR) and Multi-Agent Simulation for the SOCial sciences (MAS-SOC) [230, 31]. AOR provides extensions for modelling cognitive agents to support agent-based discrete event simulation. MAS-SOC provides a framework that allows the creation of multi-agent simulation tasks.

Even though these given systems are proposed specifically for developing agent based systems; there are some other applications which are developed by using general purpose programming languages. Those applications are mainly developed by using C++, Java, Lisp and Prolog programming languages [240].

While implementing the ideas presented by the theorists in the previous subsections, several software tools and programming languages are available. In accordance with the design purpose, an appropriate software tool or programming language can be utilized.

Chapter 3

Agent Architectures

Based on the theories presented in the previous chapter, researchers look for ways to develop agents that simulate intelligent behaviour. For this purpose, particular methodologies such as agent architectures are used. Maes [152] defined an agent architecture as a particular methodology for building agents. She stated that agent architecture should be a set of modules. Kaelbling [123] presented similar point of view and stated that agent architecture is a specific collection of modules and there must be arrows to indicate data flow among modules. In this context, agent architecture can be considered as a methodology for designing particular modular decompositions for tasks of the agents.

In accordance with these definitions, many agent architectures have been developed to simulate intelligent behaviour. These architectures are mainly categorized in three groups:

- Deliberative,
- Reactive, and
- Hybrid.

Chronologically, the deliberative architecture is the pioneering approach. Reactive architectures have been developed later in order to overcome the obstacles of deliberative architectures. Neither deliberative nor reactive architectures are able to provide good enough solutions to the real world problems. On one hand, the purely reactive architectures were not as strong as deliberative architectures in developing plans and making decision. On the other hand, the deliberative architectures was not capable of reacting to the events in their environment without complex reasoning [240]. Therefore, hybrid architectures are proposed to combine the strengths of deliberative and reactive architectures.

Deliberative, reactive and hybrid architectures are mainly concerned with simulating some aspects of intelligent behaviour. In addition to these architectures, there are also cognitive architectures. The cognitive architectures attempts to mimic certain cognitive systems like human-beings. In the following sections existing the deliberative, reactive, hybrid and cognitive architectures are reviewed.

3.1 Deliberative Architectures

In the beginning, knowledge based systems are put forward to represent intelligent behaviour on machines. Physical symbol system hypothesis is formulated to combine and form structures, and operate on symbols [170]. These types of systems in practice are disembodied constitutions that ask questions, give answers to these questions, make decisions and give advice. To operate, they require a knowledge base that is defined at the phase of development. In this manner, these systems are said to mimic intelligent behaviour.

The first symbolic agent architecture is called State Operator And Result (SOAR). Actually, it is a cognitive architecture proposed by Laird et. al. [134]. The main goal of SOAR is to achieve general intelligence. The SOAR architecture is designed according to the physical symbol system hypothesis.

The cognition underlying SOAR is tied to the psychological theory expressed by Newell [168]. This theory is called Unified Theories of Cognition. This theory attempts to explain the following questions related to intelligent entities:

- How to flexibly react,
- How to exhibit goal-directed behaviour,
- How to acquire goals rationally, and
- How to represent knowledge and learning.

Inspired from the physical symbol system hypothesis, deliberative agents are put forward. These architectures contain an explicitly represented symbolic model of the world. By symbolic manipulations and pattern matching, the agents reason to decide their actions. Deliberative agents make decisions purely based on logical reasoning [202].

Planning systems are the first instance of deliberative architectures [194]. In the AI domain, the term planning stands for the task of coming up with a sequence of actions that will achieve a particular goal. Initial planning systems adopted strong assumptions to investigate planning paradigms:

- There is only one agent which is a planner such that it can affect the world,
- While the planner agent is planning, it has a well defined goal which remains fixed,
- The planner agent has complete and accurate knowledge on the current situation,
- The planner agent possesses an accurate model of the world, and

- The planner agent has the required resources to use its model of the world to reason on the possible worlds (i.e. possible alternative courses of actions) to achieve its goal.

To illustrate these ideas, state space search techniques are applied as a planning approach. Here the term state means a possible situation that could arise in the environment. The state could represent the position of a robot or location of the tiles in an environment. The term state space implies all possible situations that could arise [140]. In state space search, the successive state of an instance is considered as achieving a goal state with a desired property. Problems are usually modelled as a state space and a set of states. The states are connected with an operation that can be performed to transform the first state into the second.

The first system that employed the state space search is called General Problem Solver which is proposed by Newell and Simon [169]. In their study, they put forward Means-Ends Analysis (MEA) to control the state space search. In MEA, when the current state and the goal state is given, an action is chosen which reduces the difference between the current state and the goal state. Afterwards, the chosen action is performed to achieve the goal state. This type of systems is called linear planner.

One of the pioneering linear planning systems is denominated as STanford Research Institute Problem Solver (STRIPS) which is developed by Fikes and Nilsson [86]. This system takes the symbolic desired goal state, the set of actions and the definition of the real world. The set of actions cover pre- and post- conditions of the actions. According to the pre- conditions, one of the deterministically defined post-conditions is chosen by an agent to achieve its desired goal state by matching the condition and the desired state. In this manner, the agent acts by using the simple Means-Ends Analysis.

This type of planning systems searches all of the possible alternatives and makes decision. After the decision making, they take the action to achieve the desired goal state. These systems are criticised as not being efficient. To operate planning systems more efficiently, Sacerdoti [195, 196] put forward the hierarchical and the non-linear planning approaches.

Hierarchical planning approaches use abstraction to reduce the complexity of the search. They divide up a problem into smaller sub-problems. Given the space and the abstraction spaces (i.e. a hierarchy of abstractions), an agent (or the problem solver) solves the problem in an abstract space. Then it uses abstract solutions as a guide to search for a solution in more detailed spaces. The hierarchical planning which uses abstraction is an effective approach but it is reported that finding a good abstraction is a very difficult task [130].

The main idea behind the non-linear planning is that a plan may have the structure of a partial ordering. The partial ordering provides some plan steps to achieve goals. Classical linear planning approaches attempt to provide a total order of plans. The non-linear planners are sometimes called the least commitment planners. In general, it means that an agent should make low commitment choices before making high commitment choices [156].

Chapman [55] criticised planning systems by presenting some theoretical results. He stated that the planning systems are not usable in the time-constrained real life situations. He underlined the fact that even if Sacerdoti's refined techniques are used, in real life situations planning systems cannot be used efficiently.

Even though Chapman's critique had been quite influential, some planning systems were proposed afterwards. Integrated Planning, Execution and Monitoring is the first of such systems and it is based on a non-linear planner [8]. Another such system is PHEONIX that is proposed by Cohen et. al. [58]. This system employs several planning agents that are designed to operate in a simulated forest fire management domain. One another system that employs planning agents is Agents in a Simulated Driving World (AUTODRIVE) which is put forward by Wood [235]. These agents are used to simulate traffic flow which is a highly dynamic environment.

In the mid 1990s, AI Planning community started investigating plan evaluation metrics to guide the search behaviour of various planning systems to overcome the obstacles stated by Chapman. The development of Planning Domain Definition Language (PDDL) influenced this interest and resulted in planning actions with duration and modelling resource consumption. The modelling time and resources allows metrics to be developed to operate planning systems more efficiently [87].

In 2002, Hawes [110] put forward an anytime planning agent to be used in computer games. He criticised traditional planning approaches and stated that they fail in a computer game environment. Based on this critique, he introduced an anytime agent which is capable of producing intelligent behaviour in a computer game environment with the hierarchical task network planner. Anytime agents form their plans iteratively and they have an additional constraint to form their plans. These constraints are called time slices and the iterations are constrained with a time slice. When the time slice runs out the best available action is taken.

Coddington and Luck [57] criticised planning systems as being disembodied. They stated that these systems are not situated in an environment, thus they lose information concerning the system and the environment. In their paper, they argued such information may be very valuable constraining plan formulation. They underlined that the context is important; since, it constraints and prioritises the goals and the actions.

With this respect, they introduced a framework for planning and plan execution. This framework can be considered as a dynamic system in which an agent generates goals in response to its motivations. According to this framework, whenever a goal is chosen, it is passed to the planner to generate a plan to achieve the goal. The planner generates a search space of alternatives during planning and it includes a plan evaluation metric to select the optimum plan for further refinement. After the decision is made, the plans are executed and updated accordingly.

Their system is also capable of getting feedback from the environment; since, they try to develop a situated system. The system evaluates the outcomes of the actions. If outcomes significantly differ from the predicted outcome; it attempts to repair the plans. Additionally, they provide flexibility to their system by enabling an agent to update its motivations which in turn may cause generating new goals or updating existing goals.

At the same period of time, symbolic artificial intelligence community spent much effort on constructing agents that are deliberative. One of these agent architectures is called Intelligent Resource-Bounded Machine Architecture (IRMA) [36]. This architecture is based on belief, desire and intention approach. IRMA has four main symbolic data structures: plan library and representations of beliefs, desires and intentions. In this architecture, there are some other components: a reasoner, a means-ends analyser and an opportunity analyser. The reasoning mechanism stands for reasoning on the world where the agent is situated. The means-ends analyser determines the plans which have potential to achieve the intentions. The opportunity analyser monitors the environment to determine further options which may be available for the agent.

In addition to these components, there are also two processes: deliberation and filtering. The deliberation process enables an agent to choose an option among the alternatives. The filtering process determines the potential courses of action and guarantees a course of actions consistent with the agent's intentions. The filtering also enables the agent to act not only resource bounded but also knowledge bounded.

The architecture is evaluated in an experimental scenario known as Tileworld by Pollack and Ringuette [182]. In their study, they investigated the behaviour of various meta-level reasoning strategies and evaluated them in a different environment. The settings of the agent and the environment was highly parameterized which in turn enabled them to evaluate different reasoning strategies by following IRMA.

Another well known instance of deliberative architectures is proposed by Vere and Bickmore [226]. This was a basic agent to operate a submarine with whom a human could interact in natural language in a simulated SeaWorld. They called this system HOMER. It could communicate by using a vocabulary which contains around 800 English words.

The agent was able to form future plans and generate new plans in response to the new information it obtained. In this system, the verbs were represented in state transition semantics for compatibility with its planner. The agent was able to give answers related to its past experiences, present activities and future intentions.

Jennings [118] put forward a layered deliberative architecture called GRATE in 1993. The behaviours of an agent that is developed according to this architecture are guided by mental attitudes including not only beliefs, desires, and intentions but also joint intentions. These mental notions were playing a central role in guiding the behaviour of an individual agent and multiple agents for solving problems. This architecture has been illustrated in a real-world domain for an electricity transportation management.

There are mainly two difficulties while developing deliberative agents: (1) Transduction Problem; and (2) Representation Problem. The transduction problem implies the difficulties in translating the real world into an accurate, adequate symbolic description. The representation problem is related to symbolically representing information about the real-world entities and processes. Based on the representation developing useful reasoning mechanism is also problematic.

Even though, deliberative architectures overcome these problems somehow, they are criticized as not applicable in practical real world situations due to searching all alternatives in a time-constraint situation. However, the deliberative architectures provide very good infrastructure for developing plans alternatives and making decisions. Therefore, these aspects of the deliberative architectures are usually merged into hybrid architectures.

3.2 Reactive Architectures

Rodney Brooks [39, 40, 41, 42] is the first researcher who criticised traditional symbolic Artificial Intelligence. He stated that the real intelligence does not have explicit symbolic representations and does not reason according to explicit abstract reasoning. Therefore, he proposed that intelligence can be generated without explicit representations and abstract reasoning. In addition, he underlined that intelligence is an emergent property of certain complex systems including human-beings.

The ideas of Brooks are based on the following observations: (a) Real intelligence is not disembodied; and (b) Interactions between intelligent entities and their environment constitutes intelligent behaviour. The reactive architectures can simply be defined as architectures that do not include symbolic world model, and do not use symbolic reasoning.

In accordance with these critics and ideas, he put forward the first reactive architecture which is called subsumption architecture. The subsumption architecture is a way of decomposing complicated intelligent behaviour into many simple behaviour modules. Each module is organised into layers which implements a particular goal of an agent. The lowest layers represent more primitive behaviours while higher layers represent more abstract behaviours. The goal of each layer subsumes those of the underlying layers. Such systems get feedback from the past decisions and perform actions accordingly. In those systems interactions between behaviours determine the actions that are going to be performed.

In contrast to more traditional architectures, subsumption architecture uses bottom-up design. Each of the layers can access data coming from sensors and the layers can give commands to the actuators to perform some actions. The main advantage of this architecture is modularity. However, it cannot support too many layers; since, the goals may interfere with each other.

He proposed the architecture for controlling mobile robots. A mobile robot is proposed to be used for wandering around and building maps of its environment. As an example, if a mobile robot developed to explore certain area and to build a map of that area, the uppermost layer would have an ultimate goal to create a map of the area. In such a case, the lowest layer could be designed to avoid objects in the area and the upper layer of it would have a goal to wander around. The lower layers of such systems would work like fast-adapting mechanisms to obtain input data while the higher layers would control the main course of action to achieve the overall goal.

The subsumption architecture is adopted in some applications. The situation and action rules are utilized for mapping in these applications. The current state of an agent determines the actions that are going to be taken; then, the agent performs actions based on the current information and it has no information related to the past knowledge [240].

At the same period of time, Agre [4] came face to face with the fact that everyday activities are routine such that they require little or no new abstract reasoning. He stated that most activities, once learned, can be accomplished with little variation. He suggested that an efficient agent architecture should be based on dynamic theories. With this respect, he proposed to develop low-level structures which only need periodic updates. He stated that if required such structures should also be capable of handling new kinds of problems.

Based on these ideas, Agre and Chapman [5] started researching on alternatives to the AI planning paradigm. They illustrated their ideas with a system called PENGI which is developed to simulate Pengo. Pengo is a computer game in which a player takes the role of a red penguin. Red penguin fights the blob-like Sno-Bees who patrol a maze. The objective of this game is to survive as long as possible by eliminating the Sno-Bees.

Afterwards, Chapman [56] developed a similar system called as Sonja. Sonja had a special-purpose function which must balance all considerations to compute a good action in any situation. For this purpose, Sonja was accessing internal data structures of the computer game it plays.

When Chapman was presenting Sonja, Woodfill and Zabih [236] put forward a new architecture for action with concentration. They presented a model of attention motivated by constraints which are psychological and computational. The architecture called Flox is designed to use their model of attention. They implemented this architecture to play Pengo. In their implementation they showed the success of autonomous systems with realistic perceptual components.

In 1986, Kaelbling [123] presented issues related with resource-bounded rational agents. Then Kaelbling proposed an agent architecture similar to the subsumption architecture. It is reported that such kind of reactive robotic control systems produced impressive results in the area of generating intelligent robotic action [183].

Following the subsumption architecture, diverse reactive architectures have been proposed by researchers in due course. Rosenschein and Kaelbling [193] specified agents in declarative terms based on their situated automata paradigm in which the agents are specified in terms of logic of knowledge.

In their study, Rosenschein and Kaelbling specified the agents in terms of perception and action components. The action component is called GAPPS while the perception component is called RULER. RULER includes three components: specifications of inputs, a set of static facts and specifications of the state transitions of the world. The GAPPS takes inputs as a set of goal reduction rules and generates a program to achieve its goals. This system actually is not purely reactive; since, it includes deliberative components.

Another successor of the subsumption architecture called Agent Network Architecture (ANA) is proposed by Maes [152]. According to ANA, an agent consists of a distributed set of competence modules which are linked in a network. These modules resemble the behaviour of the subsumption architecture. Based on the current context, an activation process which implements a competition among modules for activation energy operates on the network to determine the relevance or relative strength of a competence module.

The module with the highest activation level has the most influence on the behaviour of an agent. Learning is the core of the architecture and it is an integrated feature. The competence module network is developed and changed based on the experience of the system. Each module is specified in terms of the pre- and post-conditions and the modules are linked in accordance with these conditions. The links are added and removed depending on the observations while new macro modules are created whenever a goal is achieved.

Another purely reactive agent architecture is proposed by Boella and Damiano [27]. They stressed that the existence of implicit or explicit norms is one of the distinctive features of social systems like agents and human-beings. Therefore, they suggested incorporating explicit normative reasoning in their architecture.

The architecture is composed of three modules: deliberation, execution and sensing modules. The internal state of the agent is defined by its beliefs on the world, its goals, and its intentions. Deliberation is based on a goal of maximizing utility based on a set of preferences which are encoded in a utility function.

In this architecture, the intentions are dynamic which means they can change over time. To represent such a structure, the goal-level and the action-level commitments are introduced. In this manner, the intentions are stored in the plans which represent the goal-level commitment and the action-executions which represent the action-level commitment.

The behaviour of the agent is controlled by an execution-sensing with a meta-level deliberation. Initially, the deliberation module is invoked and the goals are matched against the plan scheme which is stored in a plan library. Afterwards, the best plan is selected as the current intention and it is executed. While executing actions, the sensing module monitors the outcome of the action and updates the world representation. If the outcome meets the agent's expectation no further action is required; otherwise, the agent undertakes re-deliberation.

These reactive architectures do not employ models of their environment. Only current state and interactions define decisions and actions. Therefore, performing appropriate actions is not possible in each time. Moreover, learning from the experiences cannot be simulated on those agents. Therefore, problems can be encountered with these types of agents while they perform their duties [240]. Despite these disadvantages, the reactive architectures are much simpler than the deliberative architectures.

3.3 Hybrid Architectures

In the 1990s, many researchers asserted that neither reactive nor deliberative architectures are suitable for real world problems. According to this assertion, researchers started studying on hybrid architectures that combine deliberative and reactive architectures. Hybrid architectures have a layered structure like the subsumption architecture.

According to hybrid architectures, agents are modelled as a set of at least two components: a deliberative and a reactive layer. The deliberative component contains a symbolic world model to develop plans and make decisions. The reactive component performs actions in an environment without complex reasoning [240].

The layers of the subsumption architecture are vertically arranged. However, hybrid architectures can be arranged either vertically or horizontally. In the horizontal layering perception is performed by every layer and the actions are controlled by every layer. In the vertical layering the reactive layer is at the lowest level of the hierarchy. While the middle level of the architecture deals with the knowledge level view of the environment of an agent, the uppermost level of the architecture represents the most abstract view.

There are many successful applications of hybrid architectures. One of the pioneering hybrid architectures is Procedural Reasoning System (PRS) [96] The illustration of PRS is given in Figure 3.1. PRS is based on the notion of Belief, Desire and Intention. It includes a plan library and explicit representations of beliefs, desires and intentions. The beliefs are the facts on the internal state of the agent and the environment where the agent is situated. The beliefs are expressed in classical first-order logic. The desires are represented as the agent behaviours and the desires are not static goal states.

The plan library contains a set of plans which are called knowledge areas. Each of these knowledge areas is associated with an invocation condition which determines when to activate a particular knowledge area. The knowledge areas can be activated in a goal- or data-driven fashion. The active knowledge areas represent the intentions of an agent. The other component of PRS is called the system interpreter and it updates beliefs, invokes the knowledge areas and executes the actions.

The Procedural Reasoning System architecture is implemented in several applications. One of the most significant implementations is called as PRS-CL architecture [165]. This research is undertaken by the Stanford Research Institute (SRI) International to represent and use procedural knowledge of experts for accomplishing goals and tasks. PRS-CL consists of four components. The first one is a database which contains current facts and beliefs. The second component is a set of goals to be

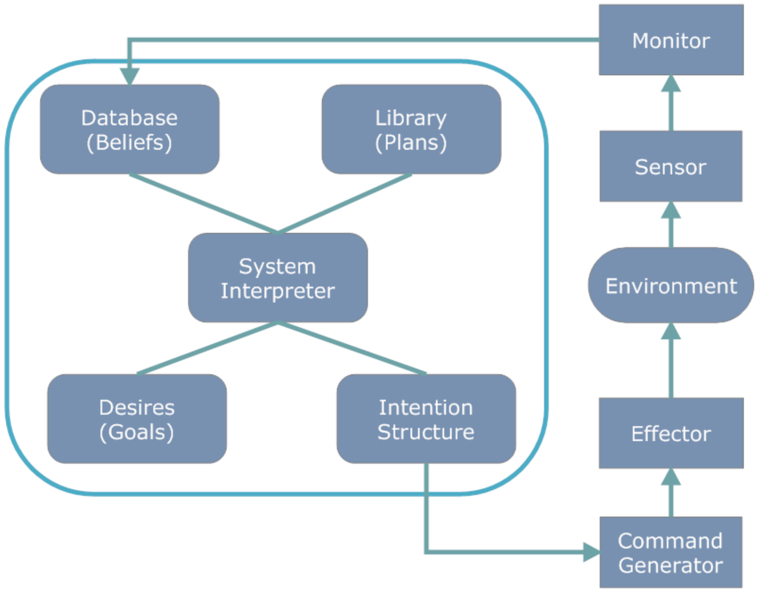

Figure 3.1: Procedural Reasoning System [Source: Lee et. al. [142]]

achieved. The third component is a set of plans or procedures which describe the details of how to achieve goals under certain situations. The last component is an interpreter which manipulates the other components to select and execute plans to achieve the objectives of the system.

Another implementation is provided by several researchers from the University of Michigan and their implementation is called UM-PRS architecture [142]. UM-PRS is an object oriented implementation of the PRS concepts by using the C++ programming language. In this research, several enhancements and simplifications in PRS are reported. These developments enabled developers to implement PRS in some application areas like unmanned vehicles.

A few years later PRS is further extended. The extended architecture is called distributed Multi-Agent Reasoning System (dMARS) [73, 74]. Both PRS and its successor dMARS are examples of the Belief, Desire, and Intention approach. In dMARS agents, the BDI approach is operationalised by plans and each agent has a plan library. The plan library of the agent represents its procedural knowledge or know-how which is related with bringing about states of affairs.

Each plan has several components: an invocation condition, a pre-condition, a maintenance condition, a body, an event queue and an interpreter. The invocation conditions which are usually specified in terms of events specify the conditions under which the plan should be considered. The pre-conditions define the circumstances under which the plan execution starts. The plans may have a maintenance condition which indicates the circumstances that must remain unchanged while the plan is executing. The plans also have a body which defines the course of action which consists of goals and primitive actions. There is also an event queue where perceived events are placed in.

The last component of dMARS is an interpreter which is responsible for managing the operations of the agent. The interpreter continually executes certain operations for monitoring, generating new desires, matching plans, managing sub-goals, and so on. One significant merit of dMARS agents is that they monitor their internal state and environment.

Another hybrid architecture is TouringMachines [84]. It has three control layers and perception and action subsystems. The subsystems of action and perception interface with the environment. The control layers perform the controlling function and mediate between the layers. There are three layers which are called reactive, planning and modelling layers.

The reactive layer is implemented as a set of situation-action rules to generate potential courses of actions for the other layers in response to the events occurring in the environment. The reactive layer is designed in accordance with the subsumption architecture.

In the TouringMachines architecture, the planning layer is responsible for forming plans and selecting actions to achieve goals. There are two components in the planning layer: a planner and a focus of attention mechanism. The planner uses a library of elaborated plans with a topological world map. By utilizing the topological world map the agent constructs plans to achieve its main goal. The planner also executes generated plans. The focus of attention mechanism filters out irrelevant information from the environment. In this manner, it reduces the amount of data handled by the planner.

The last layer is called the modelling layer and it contains symbolic representations of the cognitive state of the other entities in the environment. These cognitive states are called models and these models are manipulated to identify and resolve goal conflicts between the agent and the other entities.

Activities are produced by each layer independently and the layers communicate with each other via message passing. These layers are embedded in a control framework. This framework mediates between the layers by using control rules. It enables the agent to deal with conflicting action proposals from the different layers.

Another instance of hybrid architectures is called COSY which is a BDI architecture [43, 105]. The main components of the architecture are sensors, actuators, communications, cognition and intention. The first three components undertake the most concrete activities: the sensors receive perceptual input, the actuators ensure to perform the actions and the communications send messages.

The control elements like long term goals, attitudes and responsibilities are covered by the intention component. These control elements take part in the reasoning and decision making components. The cognition component mediates between intentions, reasons, makes decisions by choosing actions to perform.

In the cognition component, there is a knowledge base which contains the beliefs and three procedural components. These components are a script execution, a protocol execution, reasoning, deciding component, and reacting components. In COSY, the script is a plan for achieving a goal.

An agenda which contains some active scripts is obtained by the reasoning, deciding and reacting component. These scripts can be invoked in a goal- or data-driven fashion. Goal-driven fashion means that scripts are activated to satisfy one of the intentions of the agent while data-driven fashion means that scripts are activated in response to the current situation of the agent. There is also a filter in the architecture, to choose between the competing scripts.

The pioneering believable agent architecture is called Tok [23]. Tok agents are capable of reactive, social and goal-directed behaviour, and employ emotions. The Tok architecture also has natural language processing capabilities. Several successful applications of Tok have been developed.

InteRRaP is a layered hybrid architecture [162, 163]. The higher layers of the architecture represent higher level of abstraction. From bottom to top a world interface component, a behaviour-based component, a plan-based component and a cooperation component are situated hierarchically. Each layer of InteRRaP is subdivided into two vertical layers. The first one contains layers of knowledge bases and the other contains control components that interact with the knowledge bases.

At the lowest level, the world interface control component deals with acting, communicating and perception by utilizing corresponding world model knowledge base. The behaviour-based component implements and controls the basic reactive capabilities of the agent while manipulating a set of patterns of behaviour (PoB). The PoB has a STRIPS type structure. The pre-conditions define when the PoB is to be activated and post-conditions imply conditions that define the circumstances under which the PoB is succeeded or failed. There is also an executable body in PoB for defining what action should be preformed if the PoB is executed. These actions can either be primitive or planned. Primitive actions involve calling the world interface component while planned actions require calling higher layers to generate plans.

Above the behaviour-based component, there is a plan-based component. This component contains a planner to generate plans in response to requests from the lower layer. The knowledge-layer of this component contains a set of plans with a plan library. The highest layer of InteRRaP is the cooperation component. The cooperation component generates joint plans in response to the requests from the plan-based component to satisfy the goals of multiple agents. To do so the component elaborates plans selected from the plan library.

Control in InteRRaP is both in data- and goal-driven fashion. The perceptual inputs can result in a change to the world model and PoB may be activated, dropped, or executed accordingly. As a result of the PoB execution, the higher layer may be asked to generate plans and joint plans to achieve the goals. As a result of these actions, messages are generated by the world interface.

In 1993, Beaudoin and Sloman [25] proposed that while explaining autonomous agents like human-beings, it is required to account for a number of features concerned with attention and motivation. They suggested that an agent should have multiple independent sources of motivation operating asynchronously. They added that these motivations are triggered by some internal and external events like hunger and seeing a friend in trouble. They underlined the importance of attention by stating that attention is directed to meet a subset of current needs.

Based on these ideas, Beaudoin [24] put forward a new architecture. Today, this architecture is called as Motive Processing Architecture (MPA) which is another instance of PRS. There are three additional components in MPA: goal generactivation, goal management and goal meta-management.

The goal generactivation is a process for monitoring the beliefs of an agent and generating and activating goals on the basis of its desires. Once a goal is generated it is activated at the same time automatically. The goal activation makes the goal control state a candidate for directing management processes. There is also busyness filter to check the load on the planning processes of the agent. Once a goal is activated the busyness filter controls if the planning processes are busy, if not the goal passes through the filter. In MPA, if a goal passes through planning process it means that the goal is surfaced. In this manner, MPA attempts to direct and limit reasoning attention.

In MPA, goal management refers to the processes involved in making decisions on goals and/or management processes. Decision-making is the main function of the management process. To take decisions, the system performs various auxiliary functions like information gathering about the attributes of particular goals. The management process is also concerned with the control of action and management of the management processes.

The heuristic meta-management is developed in MPA to overcome the control problem stated by Hayes-Roth [111]. The control problem is related to deciding actions that are going to be performed next in good computational time. For this purpose, the meta-management controls every lower level action. When the problems or opportunities are detected, the heuristic meta-management is invoked to determine if an action should be taken in response to these problems or opportunities.

Another hybrid architecture concerning the control problem is Adaptive Intelligent System (AIS) [112]. AIS is a blackboard-based architecture which uses dynamic control plans. These plans guide the meta-level decisions which mean that the agent decides what goals to focus attention upon. An AIS agent dynamically constructs explicit control plans. These plans guide the agent to choose among the situation-triggered behaviours.

AIS is a hierarchical architecture which has mainly three processes: a perception process, an action process, and a cognition process. The perception component acquires, abstracts, and filters observed data, then sends those data to other components. The action system controls the execution of external actions that are performed by the effectors. Through the perception-action coordination processes, the perception component can directly influence the actions.

In AIS, the cognition component is realized as blackboard architecture. The blackboard is used by the agents to communicate by simply writing on a shared data structure [79]. The blackboard architectures mainly consist of three major components: the software specialist modules, the blackboard, and the control shell. The software specialist modules are the knowledge sources of the system which provide specific expertise needed by the application. The blackboard is a data structure which is a shared repository of the problems. Finally, the control shell is the controlling centre of the flow of the activities related with solving the problems.

In AIS, the blackboard architecture is extended to support the dynamic control planning. The cognition system interprets the inputs, solves the problems, makes the plans and guides the agent in the perception and the action. These main three processes operate concurrently and asynchronously while communicating by message passing.

In 1995, Luck and D'Inverno [150] attempted to specify the relation between the autonomy and the agents. They criticised that the so called autonomous agents are only objects with goals. They suggested that autonomous agents should have motivations in such a manner that they should enable them to generate goals.

Following Luck and D'Inverno, Norman [172] asserted that for an agent to be autonomous, it is necessary condition to be able to generate goals. In the light of these ideas, Norman put forward a model of a goal autonomous agent. According to his model an agent generates goals in response to the unexpected changes in the environment.

Based on this model, Norman presented an abstract agent architecture called Motivated Agency (MA). In his study, he proposed that motives cause an agent to act. Motives are used to generate motivated goals which are goals associated with some motivation. The motivations are functions to be used to evaluate the intensity level which is the motivation level. According to this architecture, the agents pursue goals with a higher motivation level first.

A motivated agent performs two important functions: goal generation and goal activation. The agent is driven by a number of motives that have the capacity to generate goals in response to the changes observed in the environment. The goals are activated under two conditions:

- if the intensity of a motivation related with a goal is sufficient to exceed a threshold, or
- if an agent decides to act on a goal.

In this architecture, there is an alarm processing mechanism which serves to focus the planning attention on the most salient goals to avoid unnecessary reasoning thereby prevent the cognitive overload. The alarms are generated on the basis of prediction of relevancy and importance of the goals. Each goal is associated with an alarm which has intensity. The threshold which is a part of alarm processing mechanisms is used for controlling goal activation. If the intensity of the alarm associated with a goal exceeds the threshold; this is called alarm triggering. Once the alarm is triggered, the associated goal is considered to be activated. When a goal is activated, the agent starts acting to achieve that goal.

In 1999, Huber [116] proposed a hybrid agent architecture called as JAM which is another instance of the BDI architecture. JAM combines BDI theories; PRS specifications; the structured circuit semantics; and the act plan interlingua. The structured Circuit Semantics is semantics of robotic languages that is used to represent the control behaviour of control systems [142]. The act plan interlingua enables a representation for creating and manipulating acts according to the plans [166].

A JAM agent is composed of five components: a world model, a plan library, an interpreter, an intention structure, and an observer. Except for the observer, the constituents of JAM are standard PRS components. The world model is a database which represents the beliefs of an agent. The plan library is a collection of plans which can be used to achieve goals. The interpreter is the component that is responsible for reasoning on what to do, when to do it and how to do it. The intention structure is a model of an agent's current goals and it keeps track of progress and accomplishment.

The observer is an optional declarative procedure which is executed by the interpreter. JAM interpreter executes observer between each action in a plan. The observer procedure is a plan with the procedural body of a plan. The programmer can use the observer to provide asynchronous capabilities. One of the important attribute of JAM agents is mobility which enables developers to building applications requiring mobility by using check point capabilities. The check points are used to save the agent's state and restore later on the same or the different environments.

McCauley and Franklin [158] proposed agent architecture called Conscious Mattie (CMattie). CMattie has the ability to display adaptive emotional states. CMattie is capable of learning more complicated emotions to interact in more complicated situations. CMattie attempts to achieve its goals reinforced by the emotional worth of them. The CMattie architecture has many capabilities like learning, being social and flexible.

In the same year, Camurri and Coglio [44] proposed an architecture for affect display. By this architecture, they were not trying to model human like agents. Instead, they were trying to illustrate architecture for affect display by providing practical behaviours. The proposed architecture leads to the agents that are social, flexible and situated.

Based upon the split of multi agent systems, M-Agent architecture is proposed by Cetnarowicz and Nawarecki [52]. In their model, every agent is characterized by an agent's mind which is composed of strategies, goals and models of the environment. The strategies provide the means to modify the models of agents. The goals represent the objectives to be satisfied.

According to their approach, the agent observes the environment and builds a model of it by using an imagination operation. While operating the imagination, the agent utilizes its strategies to obtain an estimate of the environment. Afterwards, the agent compares the environment and the model of the environment by using its goal function. The goal function determines the objective of the agent functioning. Then the agent chooses the best available strategy to achieve its goal, finally realises it. In the model, realisation is performed by the execution process. In M-Agent architecture, the decisions are taken by using a common decision function. This function is used for selecting the best strategy for execution.

In 2001, Sadio et. al. [197] proposed emotion-based agent architecture. This architecture is an advanced version of DARE which is proposed by Maças et. al. [151]. This architecture employs two types of emotions: primary and secondary emotions. In this architecture, the goals are generated from an agent's behaviours and needs.

Baillie and Lukose [18] introduced an agent architecture in which the affective decisions are made through an emotion appraisal. The architecture is called Emotionally Motivated Artificial Intelligence (EMAI). EMAI enables agents to change their behaviours based on their emotional states in guide of the interactions with the environment. EMAI agents are capable of predicting future emotional states and deciding how to behave in case of a change in their emotional state. EMAI includes a motivational drive generation component. However, EMAI only processes internal sensory data. Therefore, it does not employ motives to generate goals.

In 2005, Imbert and de Antonio [117] proposed an emotionally-oriented architecture called COGNITIVA. This architecture distinctively differs from the conventional architectures which employ an emotion component. This architecture includes three layers which are the reactive, the deliberative and the social layer. The components and the processes of COGNITIVA are designed to deal with emotions. This architecture provides some mechanisms and structures to build agents with emotionally influenced behaviour.

Originally, BDI agent architectures do not employ emotional states. Pereira et. al. [179] proposed an approach to develop emotional BDI agents. In their study, they presented a highly abstract architecture which employs an emotional component. This component enables emotions to influence the behaviour of an agent. Even though this addition is promising, this approach does not employ any motives to support strong autonomy.

In 2006, Rational Agent Architecture (RAA) is put forward by Lloyd and Sears [148]. In this architecture, agents have belief bases which can be modified by belief acquisition. Belief acquisition can be performed in two ways: by online learning and by conventional knowledge base update. Moreover, they adopt the maximum expected utility principle for selecting the rational action.

One major component is a model of the environment which is a model of enough of the environment. It enables agents to select the actions effectively. Their model has state and belief parts. They also employ a learning component in their architecture to enable agents to acquire whatever information is required for the action selection.

Karim et. al. [125] stated that one of the most important issues related with the agent research is developing an architecture that combines conventional agent approaches and learning concepts. Then, they proposed FALCON-BDI hybrid architecture. FALCON is based on a reactive learning approach which employs reinforcement learning. Reinforcement learning algorithms basically, attempt to find a policy that maps the states of the world to the actions the agent ought to take in such states.

Karim et. al. [125] underlined the fact that the reactive learning approach of FALCON and high level abstract capabilities of BDI are very promising. Therefore, they suggested combining the strengths of these approaches in their proposal. To achieve their aim, they proposed a layered architecture with FALCON at the bottom level and an instance of the BDI approach above it.

To realise their ideas, they developed uppermost layers by using JACK Intelligent Agent and the reactive learning layer by following FALCON. JACK Intelligent Agent is an agent-oriented programming language which can be used to develop BDI agents. Since it is an instance of JACK, their architecture also contains the JACK execution engine. The uppermost layer of the architecture contains a plan generation sub-component which contains a priori data and goals, a plan dispatcher and a planner. According to this hybrid architecture; plans are generated by the top-levels by using the rules learnt by the bottom-level.

Hybrid architectures have distinct advantages over purely reactive and purely deliberative architectures. It is because of the fact that hybrid architectures bring strengths of these two architectures. In this manner, the agents using hybrid architectures can generate plans by reasoning while they can act quickly.

3.4 Cognitive Architectures

While deliberative, reactive and hybrid architectures are concerned with simulating some aspects of intelligent behaviour, cognitive architectures aims to provide a blueprint for intelligent agents that act like certain cognitive systems like human-beings. Cognitive architectures propose computational processes to act like a cognitive system. These architectures form a subset of agent architectures. Cognitive architectures can be symbolic, connectionist, or hybrid. Many of cognitive architectures are based on the following idea: "Mind is like a computer" and based on a set of generic rules.

The pioneering cognitive architecture proposed is SOAR. It is a symbolic cognitive architecture. SOAR is proposed by Laird et. al. [134]. Even though, Laird et. al.

developed SOAR to achieve general intelligence, they were more concerned with the internal information processing of an agent. These processes were related to reasoning, planning, problem solving and learning.

Detailed information on SOAR is provided in Section 2.2.1 when explaining deliberative agent architectures. In addition to these, SOAR is further extended by Laird [133]. In this research study, Laird introduces non-symbolic representation to SOAR architecture. Besides, he makes major additions like new learning mechanisms and long-term memories.

Another early cognitive architecture is called Adaptive Control of Thought Rational (ACT-R) [10]. This architecture has its roots in the models of human cognition developed by Anderson [9]. The most important assumption underlying ACT-R is about human knowledge which states that the knowledge can be divided into two irreducible kinds of representations: declarative and procedural. The other important assumption of this architecture is rationality.

In ACT-R there are three main modules: a goal module, perceptual-motor modules and memory modules. The perceptual-motor modules include a simulation of the real world interface with an environment. In ACT-R these modules are usually visual and manual. There are two different memory modules called declarative and procedural memory. The declarative memory is related to the facts while procedural memory is on how to do things.

Except for the procedural knowledge, all modules can be accessed through their buffers whose contents represent the state of ACT-R at a given time. Declarative knowledge of an agent is represented in the form of vector representations of individual properties which are accessible through buffers. These representations are in some sense specialised and largely independent brain structures.

The procedural module which is used to access contents of the other modules stores and applies the procedural knowledge. The procedural knowledge is represented in a form of productions which is a formal notation to specify the information flow from the buffers.

The other component is called the pattern matcher that searches for a production and then maps the current state of the buffers at each moment. At a given time, only one production can be executed and when executed it can modify the buffers which result in a change in the state of the agent.

A decade later, Anderson and Lebiere [12] started studying neural plausibility related to the theory of ACT-R. Based on these studies a newer version of ACT-R which is version 5.0 is presented in 2002. This version introduced specialised sets of procedural and declarative representations. These could be mapped to known brain structures. Moreover, newly introduced buffers are put forward to mediate between procedural and declarative knowledge. Based on these advancements, the modified theory of mind is presented in 2004 by Anderson et. al. [11].

Afterwards, the newest version of ACT-R which includes significant improvements in coding is presented. The newest version provides unification and/or standardisation in the buffer mechanism. ACT-R 6.0 makes the system modular where a certain module can be easily added or removed. Moreover, better integration is provided in between the cognitive components and the production modules.

Another early cognitive architecture is called Entropy Reduction Engine (ERE) [76]. The ERE architecture consists of the following components: a reactor, a projector and a reductor. The reactor produces reactive behaviour in the environment. The projector is in some sense an opportunity analyser which explores possible alternatives. Afterwards, it provides advice on appropriate behaviours to the reactor. The reductor provides the means to reasoning while considering behavioural constraints. The reductor provides search and control advice to the projector.

This architecture contains long term memories which describe the effects of the actions, and the environmental and the behavioural constraints. At the same time the memories provide control rules which propose the actions to achieve goals. The long term memories are also used to generate a simple problem out of complex problems by using reduction rules. The operators and constraints of the architecture are used to produce projections which provide actions to execute. Successful projections enable an agent to learn new control rules. If an agent fails in executing actions, the agent revises its constraints and operators.

Langley et. al. [139] presented an architecture called Icarus. Their design consists of three main components: a perceptual system, a planning system and an execution system. Icarus also includes a memory system. The memory is invoked by the perceptual system and the planning system to retrieve structured experiences from the long-term memory. The long-term memory includes objects, states, and plans. The planning system uses a variant of Means-End Analysis to generate plans.

In 1992, Real-Time Control System (RCS) architecture is proposed by Albus [6]. RCS consists of a hierarchically layered set of processing modules which are connected by a network of communications. The communication system conveys messages between the modules which act as a collection of intelligent agents sending and receiving commands and requests. In RCS, there are four main modules: a behaviour generating module, a world modelling module, a sensory processing module, and a value judgement module.

The primary feature of these modules is a bandwidth of the control loops. At each layer, the bandwidth is determined by the spatial and temporal integration window of filters, the temporal frequency of events, the spatial frequency of patterns, the planning horizon, and the granularity of the planners.

The behaviour generating module is involved in job assignment, planning and control algorithms. This contains knowledge on processing tasks which includes decomposing tasks and executing them. The world modelling module covers a model of the state space of the problem domain. This module uses models to generate expectations and predict the results of the actions. The world modelling module sustains a knowledge database module which includes knowledge about entities and events. At the same time, the knowledge database, in some sense, is the memory of the overall system.

The sensory processing module receives and processes sensory inputs like visual, auditory, tactile, etc This module is involved in filtering, masking, differencing, correlating, and matching input data. Moreover, it includes recursive estimation algorithms, feature detection and pattern recognition algorithms. The value judgement module is responsible for computing cost, risk and benefit. In this manner, it generates alternative courses of action.

At the same time, Carbonell and Minton presented their cognitive architecture called PRODIGY [225]. This architecture is designed to unify problem solving, planning and learning methods. PRODIGY acts as a general problem solver by using its tightly coupled six different learning modules. The problem solver is actually a search engine that searches over a problem space defined by the operators and the environment.

The explanation-based learning module enables an agent to construct control rules based on its problem solving experience. These control rules are used to improve the search efficiency and the solution quality. In the meantime, it directs the problem solver along unexplored paths. In the absence of control rules, the problem solver searches according to depth-first Means-Ends Analysis.

Hexmoor et. al. [113] presented their architecture called Grounded Layered Architecture with Integrated Reasoning (GLAIR) in 1993. GLAIR is presented to develop cognitive robots and intelligent autonomous agents. GLAIR is a three tiered architecture with the following layers: a knowledge level, a perceptuo-motor level and a sensori-actuator level.

The perceptuo-motor level contains physical representations of objects which include object characteristics like size, weight, texture, and shape. It also undertakes unconscious acts by using routines for well-practiced behaviours. The knowledge level is also referred to as the conscious level of the architecture. In this level conscious reasoning takes place by using abstract representations of the objects. The sensori-actuator level controls operations of the sensors and the actuators. Besides, the overall architecture acts according to the intentional stance. GLAIR has been used to design and implement a cognitive robot called Cassie [201].

In the following year, a cognitive architecture called FOr the Right Reasons (FORR) is presented by Epstein [80]. FORR is put forward to model expertise at a set of related problem classes by employing learning and problem solving concepts. FORR is based upon a portrayal of the nature of human expertise and attempts to simulate it. A FORR agent can learn both from an external expert model and from its experiences in its domain.

In 1990s, Hofstadter [114] was studying on cognition. His studies resulted in many cognitive models. The most popular model of Hofstadter is called CopyCat. He considered analogy making as the core of high-level cognition and perception. He stated that high-level perception emerges from many independent processes called codelets which run in parallel, competing or cooperating. These codelets create and destroy temporary constructs to produce answers.

The codelets rely on slipnet which is a long-term memory of an agent. The slipnet is an associative network. The last component of the architecture is called coderack. The slipnet and the coderack together form a workspace which is similar to the blackboard systems. Today there are many successors of CopyCat. The most significant one is called MetaCat which is a self aware version of CopyCat [154, 153].

Kieras and Meyer [128] presented an architecture called Executive-Process Interactive Control (EPIC). By this architecture, their primary goal was to develop and validate a cognitive modelling architecture. They attempted to develop an agent architecture to accurately simulate human information processing for perceptual, cognitive and motor activities. EPIC provides a model to mimic human activities efficienly and accurately in the human-system interaction domain.

Many researchers were working on three layered architectures to reliably perform complex tasks [92]. They all came up with similar solutions which consist of three main components: a reactive feedback control mechanism, a deliberative planner, and a sequencing mechanism connected to the first two components [62, 91, 28]. Based on these ideas, a cognitive architecture called 3T was developed [137].

This architecture is also three tiered thus it is known as 3T. The reactive layer is the lowest layer and it includes a set of hardware specific situated skills. These skills represent the connection with the world where the agent situated. The term situated skills means capabilities that enable an agent to achieve or maintain a particular state in the environment. The deliberative layer provides the means for planning which involves reasoning about the goals. The sequencing mechanism is located in the upper most layer, which is connected to the other layers. This mechanism activates the situated skills to achieve specific tasks [29].

Another cognitive architecture is called New Millennium Remote Agent Architecture (NMRAA) [178]. A NMRAA agent mainly has four components: a mission manager component, a planning/scheduling component, a smart executive component, and a mode identification component and a configuration component.

The planner/scheduler that generates new mission sequences is a constraint-based planner and a resource scheduler. The planner/scheduler is activated by the mission manager when a new plan is requested by the smart executive. The mission manager formulates short-term planning problems for the planner. These short-term plans are generated based on the long-term mission profiles. The assembly and the execution of the actions are performed by the smart executive component. The last component which involves mode identification and configuration enables an agent to diagnose failures and reconfigure the hardware.

To operate in uncertain environments another cognitive architecture is put forward by Freed [88]. This architecture is called Apex. The most significant component of Apex is a reactive planner which selects actions from a library. The library is the storage which holds partial plans. Its reactive planning algorithm makes decisions on the next course of action. It is called reactive since new decisions can evolve when information relevant to a decision becomes available. Another merit of this architecture is that it can manage multiple tasks efficiently by resolving resource conflicts.

In 2000, Gratch [102] proposed a model based on emotional reasoning named Èmile. Èmile is built on prior computational models of emotion, specifically En algorithm and Affective Reasoner [77, 188]. This architecture has five processes: a planning, an appraisal, an appraisal evaluation, an integration, and a guidance process. In its current form, Èmile can learn about the activities of the other agents by observing their actions or communicating with them. In other words, it adopts social learning theory.

In the planning process of Èmile, an agent plans and manipulates plans to determine which action to be selected to pursue its goals. The appraisal process shows the relation between the plans and the goals. In the appraisal evaluation process, the agent assigns a quantity to the appraisal. Afterwards, Èmile integrates appraisals into an overall emotion state. In the final process, these appraisals guide the action selection and the planning. Like the models of emotion, the term appraisal refers to the process of qualitatively evaluating the emotional significance of events.

CHunk Hierarchy and REtrieval Structures (CHREST) is another cognitive architecture. It is proposed by Gobet et. al. [98]. It is actually the successor of the cognitive model called Elementary Perceiver and Memorizer (EPAM) that is developed by Simon and Feigenbaum [82]. The EPAM is based on a psychological theory of learning and memory to simulate verbal learning. Verbal learning refers to a learning style when an agent acquires information via written or spoken language.

The most important part of CHREST is learning which is modelled as the development of a network of connected nodes. These nodes are called chunks. CHREST contains several capacity and time parameters. The capacity parameters include capacity of the visual short-term memory and set at three chunks. The time parameters cover time to learn a chunk to put information into memory.

Another cognitive architecture is proposed by Sun et. al. [218, 216]. In 2001, they proposed Connectionist Learning with Adaptive Rule Induction ON-line (CLARION). CLARION is proposed to provide a skill learning model. The structure of this architecture is given in Figure 3.2. According to their model, the procedural knowledge is developed first and then the declarative knowledge is developed.

The architecture includes a number of subsystems: an action-centred subsystem, a non-action-centred subsystem, a motivational subsystem, and a meta-cognitive subsystem. While the action-centred subsystem controls the actions, the non-action-centred subsystem maintains the general knowledge of the agent. The meta-cognitive subsystem monitors, directs, and modifies the operations to improve the efficiency of the processes.

The motivational subsystem provides the underlying motivation for perception, action and cognition. The motivational system provides drives and defines the interactions between them. In the architecture, they define two types of drives: primary and derived drives. The primary drives include low-level and high level drives. Low-level drives are essential and built-in drives. In other words, these drives are physiological. Beyond these low-level drives, the other hardwired drives are the high-level drives. While explaining these drives, they follow Maslow's hierarchy of needs and classify

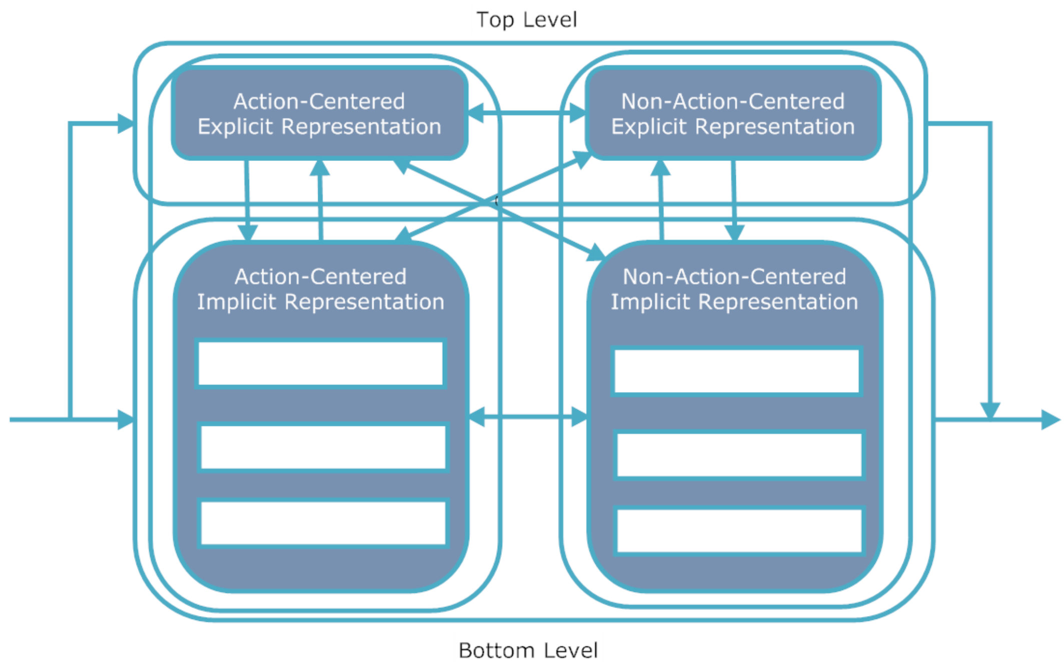

Figure 3.2: The Structure of CLARION [Source: Sun et. al. [218]]

these drives as belongingness, esteem and self-actualisation needs. In the architecture, derived drives which are the secondary drives can change over time. These drives are acquired in the process of satisfying primary drives.

At the end of the 1990s, Sloman [209] was studying on architectures to find a way to give human-like powers to agents. He criticised AI and Cognitive Science by only addressing the components of such architectures. Besides, he stated that there should not be a fixed architecture to mimic intelligence. Then he proposed the gross features of the human information processing architecture scheme. By this scheme, he suggested three control layers: a reactive, a deliberative and a meta-management. While the reactive layer is at the bottom of the architecture, the meta-management component is the top layer.

In his scheme, the reactive and the deliberative layers are in their classical form.

While the reactive layer interfaces with the environment, the deliberative layer generates plans and undertakes the decision-making processes. Besides these layers, he puts forward one additional layer called the meta-management. This layer provides self-monitoring, self-evaluation and self-control capabilities. The meta-management layer controls and monitors the deliberative layer. The deliberative and the meta-management layers also have long-term memory.

Another important attribute of Sloman's scheme is that it includes a motive activation mechanism. Motives are used as filters to focus the attention of the agent. The attention mechanism, which selects motivators to attend to, includes motivator generators, attention filters, and a dispatcher [25]. In this scheme, he interprets emotional states as arising out of the perturbances. In other words, he states that certain emotional episodes are the phenomenon of a partial or total loss of control of attention [241].

This cognitive architecture scheme is called Cognition and Affect (CogAff). Figure 3.3 shows CogAff architecture scheme. H-CogAff is an instance of the CogAff architecture scheme [211]. By H-Cogaff, Sloman gives special emphasis to emotions. Sloman first distinguishes emotions in two categories as primary and secondary like Damasio [67], Goleman [99], and Picard [180]. Later, Sloman [210] proposes tertiary emotions. The primary emotions include the most primitive emotions such as being startled, frozen with terror, sexually aroused. The secondary emotions are the emotions like apprehension and relief. The tertiary emotions are the result of peturbances and such emotions are infatuation and humiliation.

In this architecture, he relates each emotion type with a certain layer. The reactive layer accounts for primary emotions. He states that the secondary emotions require reasoning abilities; therefore, the deliberative layer supports secondary emotions. Finally, he stresses that the meta-management layer not only supports control

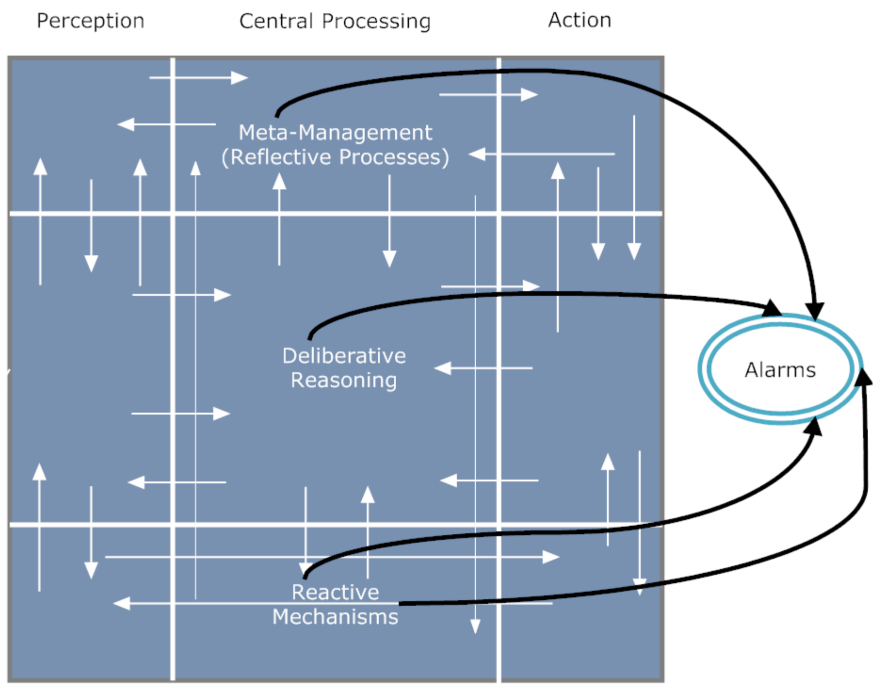

Figure 3.3: Cognition and Affect Scheme [Source: Sloman [211]]

but also supports the loss of control. With this respect, this layer is involved in generating tertiary emotions which result from the phenomenon of a partial or total loss of control of attention. Moreover, he states that the secondary and tertiary emotions are probably unique to humans.

In 2001, another cognitive architecture called Shadowboard which contains a collection of individual sub-selves (i.e. sub-agents) making up a whole agent is presented [100]. According to this architecture, nine different sub-selves constitute one single agent. The first sub-self which is called aware ego agent is in the middle of the architecture. These sub-selves have envelopes-of-capability which represent different areas of expertise. Each envelope-of-capability also contains a number of sub-agents with similar capabilities.

There is also one purely reactive sub-self with a rule-action mechanism. Another agent has deliberative capabilities to be used when the resources are available. All sub-agents have knowledge about the capabilities of sub-selves in their envelope-of-capability. They use this knowledge to select a suitable sub-agent to achieve certain goals.

Lee and Zhao [143] proposed a real time agent architecture based on the human cognition model of Neisser [167]. Neisser's model considers human cognition as a perpetual process which works as long as the human-being is awake. In his model, a human-being acquires samples which bring about useful information by exploring the environment. The human-being makes decisions and plans by modifying information. These plans and decisions guide the being to explore the environment to obtain further information. His model also rejects introspection as a valid method and explicitly acknowledges the existence of the mental states such as belief, desire and motivation.

Even though this human cognition model underlines the importance of motivation, Lee and Zhao overlook this fact in their architecture. They focus on the state of being awake and stress that human-beings are the best examples of a real time agent. With this respect, they realised real time agency through satisfying automation, reaction, real time AI, perception and selectivity attributes.

In their architecture, an agent has three subsystems: a perception, a cognition and an action subsystem. These subsystems acquire from and respond to the environment through real time AI reasoning concurrently and synchronously.

Bozinovski and Bozinovska [33] presented their computational theory of emotions. In their theory, they assume that the feelings of human-beings indicate how they evaluate people, events, and things. They consider emotion is an internal self-evaluation of something relevant for the existence of the agent. Self-evaluation can be related to the evaluation of the global state of the agent and the behaviours of the agent.

Based on these ideas, they put forward an agent architecture which contains a genetic control system, a neural component, and a hormonal component. This architecture is based on a crossbar connectionist adaptive array. It computes from the crossbar memory elements to determine emotions toward situations encountered and action tendencies. In this way, the architecture provides connections between emotions and behaviours.

Cassimatis [51] proposed a cognitive architecture called Polyscheme. Polyscheme is designed to model and achieve human-like intelligence. The architecture consists of several specialists. Each specialist models a different aspect of the world.

In Polyscheme, the specialists communicate with each other and by using the same focus of attention they execute particular operations. The focus of attention enables specialists to focus on the same aspect of the world simultaneously. The architecture also employs several inference techniques such as script matching, backtracking search, and stochastic simulation.

In 2006, Brain Inspired Cognitive Architecture (BICA) is put forward by Shanahan [199]. The architecture incorporates the concepts of consciousness, imagination, and emotion. To simulate consciousness, Shanahan adopts a model of information flow from Global Workspace Theory (GWT).

Global Workspace Theory is another cognitive architecture that contains a large set of conscious and unconscious processes [15, 16, 17]. GTW contains conscious and unconscious processes which are usually explained by the theatre metaphor of mental functioning. Consciousness is called a bright spot on the stage directed there by focus of attention under executive guidance. Consciousness is realised through a fleeting memory which is limited with work duration of a few seconds. The primary function of consciousness is to allow blackboard architecture. By using the blackboard architecture it coordinates the other specialised networks. The rest of the theatre is the unconscious part which is called behind the scenes.

By employing GWT, Shanahan claims that BICA emulates the consciousness.

The cognitive functions like anticipation and planning in BICA are realised through internal simulation of interaction with the environment. In BICA, action selection depends on an affective system. The architecture is neurologically believable; since, it contains a variety of the brain structures and systems. The overall implemented system covers four modules: a visual system, an affective system, an action selection system, and a broadcast system.

In 2008, Jilk et. al. [120] proposed a cognitive architecture called SAL (Synthesis of ACT-R and Leabra). This architecture combines ACT-R and Local, Error-driven and Associative, Biologically Realistic Algorithm (LEABRA)[173]. LEABRA is a neural architecture which models the neocortical learning mechanism. He explains this mechanism as a combination of error-driven and self-organizing learning. Based on these two architectures, SAL is realised as a hybrid symbolic-connectionist architecture.

Wang et. al. [233] proposed Layered Reference Model of the Brain (LMRB). LMRB covers thirty seven cognitive processes of natural intelligence. This model contains six layers called sensation, memory, perception, action, metacognitive, and higher cognitive layers. While the lowest layer is sensation layer, the highest layer is higher cognitive layer. The lowest four layers are categorised as subconscious life functions. The highest two layers, metacognitive and higher cognitive layers, are categorised as conscious life functions. LMRB is inspired from the structural model of the human brain.

Later, Wang [232] modelled perceptual processes like emotions, motivations, and attitudes. He stated that the emotions arise from the factors like a personal feeling and an internal status, a mood, circumstances, a historical context, and external stimuli. He classified emotions in three levels: super, basic, and sub-category. Super level categorises emotions as positive and negative. Basic level covers basic emotions like joy, love, anger, sadness, and fear. Sub-category level includes more complicated emotions each of which is related with basic emotions. In addition, Wang listed emotions hierarchically as weak, moderate, strong and strongest emotions. In his model the strength of motivation is proportional to the strength of the emotion. Furthermore, motivation is defined as a desire triggered by an emotion or external stimulus to pursue a goal. Wang [231] also proposed Cognitive Informatics Reference Model of Autonomous Agent Systems (AAS). In this study, he states that software agents are goal-driven.

In 2009, Laird et. al. [136] attempted to evaluate human-level intelligent systems. In that study they note that there is a significant increase in cognitive architecture and general intelligence research in the last five years. They stated that the primary claim in this field is that human-level intelligence can be achieved without exact reimplementation of human brain. They also listed a number of claims and attempted to evaluate these claims. As a conclusion, they stated that developing human-level intelligent systems is a huge challenge. They suggested researchers to be more explicit in their claims in order to progression in this field.

In the same year, Langley et. al. [138] analysed the research issues regarding cognitive architectures. They discussed various capabilities that a cognitive architecture can support. These capabilities are:

- Recognition and categorisation,
- Decision-making and choice,
- Perception and situation assessment,
- Prediction and monitoring,
- Problem solving and planning,
- Reasoning and belief maintenance,

- Execution and action,
- Interaction and communication, and
- Remembering, reflection, and learning.

After determining these capabilities; Langley et. al. state that there is considerable need for further research on cognitive architectures. They indicate that the most important area to be considered for further development is introduction of new capabilities to existing architectures. They underline the fact that only few support capabilities listed above.

In 2010, Laird and Wray [135] attempted to define requirements for achieving artificial general intelligence. They outlined eight characteristics regarding environments, tasks, and agents. By considering those characteristics as influences on desired agent behaviour, they derived twelve requirements for general cognitive architectures. However, they noted that many of the derived requirements are vague. Therefore, it would be difficult to apply those requirements to the existing architectures.

In some of the cognitive architectures different components can be active concurrently. Concurrency can be considered as one of the most basic requirement for robots. By supporting concurrency, cognitive architectures may have multiple sensors and effectors in complex and dynamic environments.

Many cognitive architectures composed of different kinds of sub-architectures. These sub-architectures may be merged into the other hybrid architectures to enhance their strengths further. From this point of view, it can be asserted that the hybrid and cognitive architectures are the most promising approaches to simulate human-like intelligent behaviour.

Chapter 4

Foundations of the Proposed Approach

One of the aims of this study is to establish a framework for simulating human intelligence. To achieve this objective, it is proposed to combine motivation theories and the intentional notion. In particular, by the proposed approach the theories of needs and Belief, Desire, Intention approach are brought together. By combining these approaches, agents are enabled to be driven by their motives, specifically their needs. In this manner, the emergence of intelligent behaviour is explained as the result of unsatisfied needs.

In the literature, the simulation of intelligence is tried to be achieved under certain assumptions. The most commonly accepted assumption in the literature is rationality. Instead, by this study, causality is put forward as the most basic assumption of the simulation of intelligence.

Constrained by such assumptions, researchers have attempted to explore the key attributes of intelligence. Then these key attributes are attempted to be satisfied while developing intelligent agents. In other words, these attributes are ascribed to the agents. The most commonly accepted attributes of intelligence are proposed by

Wooldridge and Jennings [240] as autonomy, situatedness, and flexibility. In addition to these, Russell and Norvig [194] proposed the learning attribute as one another key attribute.

In the present approach, these key attributes are utilized. In addition, the flexibility attribute is elaborated further by dividing this property into two different attributes. While the first one is still called flexibility, the second one is called social ability. Furthermore, by the proposed approach, three more attributes are added: employing motives, intentionality and ability to display affect. As explained, the proposed approach combines the theories of motivation and the intentional notion. This study asserts that intelligent beings are being driven by some motives and they act intentionally to satisfy those motives. Therefore, two additional attributes of intelligent beings are proposed as employing motives and intentionality.

Moreover, while trying to explain the intelligent behaviour it is inevitable to admit the importance of emotions. All of the intelligent beings are affective systems; therefore, while trying to simulate intelligence, affective aspect of intelligence should also be simulated. With this respect, ability to display affect is proposed as another key attribute of the intelligence.

In this section of the study, a general framework to simulate all aspects of intelligence is attempted to be build. Initially, the rationality assumption is analysed and criticised. Instead of rationality, causality is proposed as the most basic assumption of the intelligence simulation. Afterwards, in this section the details of the present approach are elaborated while the stated key attributes is explained. Finally, these key attributes are brought together in the last subsection.

4.1 The Underlying Assumptions

4.1.1 Rationality

Even though, the intentional notion provides a good theoretical infrastructure for agency, it has few but vital obstacles in simulating intelligence. Some of these problems are explicitly stated by the founder of these stances -Daniel Dennett [70]. He stated that there were some unknown issues related to the emergence of intelligent behaviour. Besides, some other objections related to the intentional notion can be put forward.

First of all, the animals cannot only be considered as intentional systems. In other words, too much abstraction in some cases may lead to false predictions in the behaviours of intelligent animals. In the end, these creatures -including humans- also have so called physical and design stances. The animals are bounded by the physical and the chemical principles of the universe; therefore, this situation certainly has an affect on intelligent behaviour.

Besides, the behaviours of animals are delimited by existential attributes, purposes and functions and this stance is known as the design stance. Here instead of the term design, the existential attributes are used to refer to the same thing. The term existential attributes are used; since, this term does not refer to a creator. Usually the term design brings about belief on a creator. In the present study, it is not aimed to refer to a creator. Instead it is tried to express that all intelligent creatures have existential attributes regardless of whether a supreme deity created them or not.

Secondly, the cause of the emergence of intelligent behaviour is not known as stated by Dennett [70]. It is commonly accepted fact that intelligent animals including human-beings are intentional. In other words, the behaviours of the animals can be somehow understood and predicted by employing the intentional stance. However, it cannot be known what causes the emergence of intelligent behaviour; therefore, the emergence of intelligence cannot be simulated by only applying the intentional notion.

Last but not the least objection to the intentional notion is related to the rationality assumption. While explaining these stances, Dennett [71] simply assumes that the agents are rational. The rationally debate is initially put forward by Sloman and Logan [212]. They stated that the systems that are developed by utilizing findings of AI are neither rational nor irrational.

Even though, Sloman's opposition is a good starting point a more serious objection is proposed by Stephen Stich [214]. He questioned if a man is ideally rational or irrational. He argued that the human-beings often have beliefs and/or desires which are irrational and the intentional stance does not help understanding and predicting the behaviours which are the result of irrational beliefs and/or desires.

When explaining the intentional notion, Dennett [70] stated that the animals are to be treated as rational agents and then attempted to understand what beliefs an agents ought to have, given their situation and purpose. However, as explained by Stich [214], human-beings have beliefs and desires which are irrational. Therefore, while trying to understand and predict the behaviours of intelligent beings, the intentional stance fails to explain the behaviours that are the result of irrational beliefs and/or desires.

Moreover, in many cases rational behaviour depends on time, culture, context, limited with the beliefs an intelligent-being has. Various behaviours of human-beings in the past are thought to be rational, while today some of them look irrational. Many more behaviours are considered as rational in certain cultures, but in other cultures they are presumed as irrational. Moreover, all intelligent behaviours are limited with the beliefs of an intelligent-being, which in turn may yield irrational actions.

As an example, around 5000 years ago when Sumer, the earliest civilization was in power, many people were practicing polytheistic religion with anthropomorphic gods and/or goddesses. These deities were representing forces or presences in the world [131]. At that time, nobody would question people worshipping those deities but today most of the people would say that it is irrational behaviour to idolize several gods and/or goddesses. Therefore, rational behaviour is time dependant; the rational behaviour of one time can be irrational in another time.

Moreover, rational action differs from one culture to another. As an example, men and women relationships specifically marriage can be examined. In some countries like Tibet and Saudi Arabia, it is common and very rational to marry with many spouses. However, in some other countries like Turkey, polygamy is prohibited by laws due to the ethical concerns and many other reasons [129]. As an example, in European countries if one would ask about marrying with several women at the same time, most people would say that it is an irrational thing to talk about because of the equity of women and men.

Besides, rational actions usually depend on the context. Assume that a man, say Jack, and its family are sleeping in their house; somebody breaks into their house. What should Jack do? Assume that Jack has a gun, then Jack would shoot the burglar with a high probability. But is it a rational action? Most of the people would say “Yes”. It is because of the fact that it is self-defence; since, Jack is protecting himself and his family. But as an example, in Turkey if you shoot a burglar in your house, it is a punishable crime. In Turkey, the self-defence rule applies only if one fires a weapon in its bedroom. Therefore, shooting the thief in other rooms is an irrational action to take in Turkey. As stated before, rational action depends on the context which in this case is the regulations and laws of Turkey [19].

The last but not the least concern in here is that rationality is constrained with the beliefs an intelligent-being has. For instance, consider a cat is given to Jack as a present. Assume that Jack does not know much about cats. The following day, assume that he wants to go outside. Before going out, he decides to keep the cat away from his parlour. Therefore, he rationally closes the door of the parlour!

Is it really a rational behaviour to close the doors? We would say "No" knowing that some cats can open a closed door. Cats are capable of using a door handle. The cats can jump on a handle and by pressing down; they can open unlocked doors [222]. Therefore, Jack should have locked the door which is the rational action to take before going out. But when deciding, he had a belief which is "Cats cannot open the closed doors.". Actually, this belief is not sufficient to provide the rational action. Therefore, as stated previously intelligent behaviour is limited with the beliefs intelligent-beings have.

In this study it is asserted that the rationality assumption fails to explain all aspects of the intelligent behaviour. However, without the rationality assumption the intentional notion is very useful in explaining, understanding, and predicting intelligent actions.

4.1.2 Causality

Instead of the rationality assumption, by this study it is asserted that the most basic assumption of intelligence simulation may be causality. Basically, causality denotes the relationship between one event and another event which is the consequence of the first. The first event is called as the cause while the latter is called the effect. The cause must be prior to, or at least simultaneous with, the effect. According to the causality, the cause and the effect must be connected by a nexus which is a chain of intermediate things in contact [176].

By the proposed approach, it is suggested that intelligent behaviour is produced in accordance with causality. Firstly, intelligent entities observe their internal state and the environment. Then based on the perceived input, intelligent entities perform some actions in the environment. According to this viewpoint, perceived input is the cause while the taken action is the effect.

Actually, all of the current architectures which try to mimic intelligent behaviour are in accordance with causality, even if the researchers are not aware of this fact. In other words, the existing architectures produce intelligent behaviour in accordance with causality. Therefore, the most basic assumption of the simulation intelligence is that intelligent behaviour is produced in accordance with the causality.

Aside from this fact, according to the present approach each cause and effect is connected by particular motive. In other words, in the proposed approach the nexi are the needs. These needs provide the means to select among alternatives. From this point of view, while producing intelligent behaviour the intelligent beings choose a plan which satisfies their motives (i.e. needs) best.

Within this point of view, one can understand and predict an intelligent-being's behaviour if he knows its motives. Instead of considering the rational actions an intelligent-being has, one should consider its motives. Based on these motives, acts of an intelligent being can be understood and predicted by considering the action which satisfies the associated motive best.

4.2 The Key Attributes of the Proposed Approach

4.2.1 Autonomy

One of the most defining attributes of intelligent beings is autonomy. Even if in some cases, animals can get assistance from others, they are capable of performing actions without assistance or guidance of other entities. Besides, they have control over their internal states and actions.

Based on this fact, Wooldridge and Jennings [240] stated that the one of the key attributes of intelligence is autonomy. Therefore, they proposed that the intelligent agents should be autonomous. According to their definition, the term autonomy refers to entities that can perform actions without the assistance of other entities. Moreover, these entities have the ability to control their internal state and actions.

Luck and D'Inverno [150] proposed that to have a stronger autonomy, agents should also have motivations in such a manner that motivations should enable them to generate goals. They considered agents with motivations as autonomous agents, while they considered objects with goals as agents only.

Abdelkader [2] stated that there are two interpretations of autonomy as self-governance and independence. Self-governance indicates an agent which is capable of selecting what goals have to be achieved in the guide of its motives. The independence refers to an agent that is independent from the other agents. He proposed self-governance as a sufficient condition to be autonomous.

In their study, Weigand and Dignum [234] stated that autonomy is still a poorly understood concept. However, they stated that to be considered as an agent a software system must fulfil autonomy. In addition to these, they stated that autonomy cannot be interpreted only as independence. Besides, they underlined the fact that an agent is dependent on its environment and the environment is also dependent on that agent. This means, the agent performs a certain role in an environment.

Carabelea et. al. [46] studied on classifying the different forms of autonomy. They stated that the most difficult problem in multi agent systems is to allow the agents to be autonomous while a coherent behaviour of the system is ensured. In their study, they explained the following types of autonomy:

- U-Autonomy (User-Autonomy): In this type of autonomy, an agent is independent form the user for choosing what action to perform. In some cases, this type of an autonomous agent can still pass the control of its actions to the user.

- I-Autonomy (Social-Autonomy): This type of autonomy is used to to refer the adoption of goals. I-Autonomy implies an agent that is autonomous with respect to the other agents for the adoption of a goal. The agent cannot be imposed to adopt a goal by the other agents.

- O-Autonomy (Norm-Autonomy): To reduce the degree of non-determinism caused by autonomous agents in a multi agent system, norms like social laws and conventions can be used to restrain the autonomy of the agents. In this type of autonomy, an agent is autonomous with respect to a norm.

- E-Autonomy (Environment-Autonomy): In this type of autonomy, the environment has an effect on the agent. However, the environment cannot impose what to do.

- A-Autonomy (Self-Autonomy): This form of autonomy can be seen as an attribute which gives an agent the ability to choose different behaviours among the alternatives it has.

After defining these autonomy types, Carabelea et. al. put forward an autonomy definition. Their definition is as follows:

> "An agent is autonomous with respect to the other agents for an autonomy object in a given context, if in the context, its behaviour regarding the autonomy object is not imposed by the other agent.".

Russell and Norvig [194] stated that if the actions of an agent are only based on its built-in knowledge, then the agent lacks autonomy. From this point of view, they stated that the agents should also have the ability to learn from experiences. In this manner, they put forward a stronger sense of autonomy.

The proposed approach covers the attributes learning and employing motives. By employing motives in the present approach, the agents are allowed to be capable of generating goals. In addition to these, learning in the proposed approach allows the agents to learn new plans to undertake in case of need.

4.2.2 Situatedness

All intelligent beings are capable of getting data from their environment and changing the environment. Intelligent beings observe the environment and obtain the sensory data from the environment. Based on those observations, they are capable of taking actions which may result in a change in the environment. They are physically existing beings in such a way that they can affect their environment.

Wooldridge and Jennings [240] put forward the term situatedness to explain this attribute of intelligent beings. They stated that the term situatedness implies entities that are capable of getting sensory data and performing actions to change their environment. Russell and Norvig [194] explains situatedness as a process of agent deliberation which is directly connected to an environment.

Accordingly, an agent is situated if:

- The agent exists in a dynamic environment of which state changes over time,
- The agent can manipulate or change the environment through their actions, and
- The agent can sense or perceive in the environment.

Chandrasekharan and Esfandiari [53] studied software agents and situatedness. They stated that software agents are not embodied. While explaining situatedness, they considered the behaviours of honeybees which live in colonies like ants and termites. They explained two behaviours of foraging bees as foraging in finding food and communicating location of the food with the other bees. They stated that the foraging activity does not require representation while communicating location requires representation.

Based on these explanations, they stated that there are two types of situatedness:

- The first type of situatedness is the situatedness in the word which allows the bee to forage.
- The second type of situatedness which can be called social situatedness is the situatedness among other agents.

These two types of situatedness are different from each other. The second type of situatedness involves exchanging symbolic structures in order to communicate.

The proposed approach is proposed to support the development of agents that are both situated in a dynamic environment and situated socially. Here the term dynamic environment is one in which the agents exist. Static environments are assumed to remain unchanged if agents do not take actions. If agents perform some actions; the static environment changes. In contrast to the static environment, dynamic environments has other processes operating on it; therefore, it may change even if an agent do not perform any action. With this respect, an agent does not have complete control over dynamic environment, like real world [238].

Based on these ideas, the proposed approach adopts the definition of Wooldridge and Jennings while also taking Russel and Norvig's explanation into account.

4.2.3 Employing Motives

In the view of the given examples in Section 4.1, the behaviours such as worshipping many deities, marrying with several spouses, shooting a burglar in the parlour, and closing a door of a room to keep a cat away from the room may or may not be rational but all those behaviours are somehow driven by some motives.

It can be asserted that while worshipping several deities, human-beings are motivated to achieve self-relief. In case of marrying with several spouses maybe people are motivated to provide an alternative for divorce. The motivation behind shooting the burglar is quite obvious -protection. Keeping the cat away from a parlour may be motivated by having clean rooms. Even the cat must be driven by a motive to open the door. For instance, the cat might have felt hungry and attempted to find food in the other rooms; since, it is another fact that the cats are hungry almost all the time!

As intelligent beings are interacting with the environment, they are motivated to take some actions. In other words, intelligent beings have motivations; based on these motives they act intentionally to satisfy them. The motivation is the reason behind the emergence of intelligent behaviour. The problem is to find those motives which cause them to act. Today, there are several different motivation theories which can be categorised in three groups: the drive reduction theories, the theories of needs, and the cognitive theories.

Before going into the details of these theories, it would be reasonable to explain the term motivation. Britannica Concise Encyclopaedia [78] defines the term motivation as the factors within an animal including human-beings that arouse and direct goal-oriented behaviours. More precise definition is proposed by Geen [93], who asserts that the motivation refers to the initiation, direction, intensity and persistence of the behaviour.

The other psychologists like Sigmund Freud have also studied motivation. Freud [89] proposed that all behaviours are the result of biological instincts. He categorised these instincts as life (sexual) and death (aggression). Besides, he stated that there is no difference between human and animal motivation.

On the other hand, many of the other psychologists have different viewpoints on motivation. Jung [122] proposed that the temperament and search for soul or personal meaningfulness drive human-beings. Adler [3] believed that will to power is the basic motive of human life. His ideas are not much different from Nietzsche [126]. Sullivan [215] stated that interpersonal and social relationships are fundamental in explaining human motivation.

The drive reduction theories are based on the ideas of Walter B. Cannon [45] who proposed that the basic human drives serve homeostatic functions. This function is fulfilled by directing the energies towards the reduction of the physiological tensions. There are several drive reduction theories in the literature. In general, theory states that the human-beings have certain biological or psychological drives like hunger. As time passes, the strength of the drive increases as it is not satisfied. If the derive is satisfied then the strength of the drive is reduced. As an example, a person who is hungry eats to reduce the tension caused by the hunger.

According to the drive theories, all human behaviour can be attributed to the pleasure obtained when the tensions are reduced. Today, the drive reduction theories lost favour. They can successfully explain the reduction of tension but they fail in explaining all intelligent behaviours. As an example, they fail to explain why people ride a roller coaster; since, it increases tension by causing fear and anxiety [13].

Another theory is called the theories of needs. Among many needs theories, the hierarchy of needs theory is put forward by Maslow [155] is the most commonly known approach. According to this theory, the human-beings are said to be motivated by their unsatisfied needs. Maslow lists these needs in a hierarchical order in five groups: physiological, safety, love and belongingness, esteem and self-actualization.

The lowest needs are the physiological needs. These are explained as the basic needs such as air, water, food, sleep, sex, and so on. When these needs are not satisfied sickness, irritation, pain and discomfort can be felt. These needs motivate human-beings to alleviate them as soon as possible to establish homeostasis.

The environment is usually so chaotic that the safety needs become very vital. A human-being needs security of herself and her family. This need can be satisfied with a secure house and a good neighbourhood. In addition to this, the safety needs sometimes motivate people to be religious. The religions comfort with the promise of a safe secure place after death and leave the insecurity of this world.

Love and belongingness are the next level of needs. Human-beings need to be a member of groups like family, work groups, religious groups, and so on. They need to be loved by the others and need to be accepted by the others. According to Maslow, it is essential to be needed.

The esteem needs are categorized in two as self-esteem and recognition that comes from others. Self-esteem results from the competence or the mastery of a task. The recognition is getting attention from the others. The people, who have all of their lower needs satisfied, often seek luxury expenditures because doing so raises the level of esteem.

The need for self-actualization is the need in becoming everything that one is capable of -as far as physical and mental capabilities permit. Human-beings that meet their other needs can try to maximize their potential. They can seek knowledge, peace, aesthetic experiences, self-fulfilment, and oneness with deity, etc...

According to Maslow certain lower needs are to be satisfied before the higher level needs can be satisfied. From this perspective, once the lower needs are alleviated, human-beings can think about higher level needs.

Another theory is offered by Clayton Alderfer [7]. He put forward ERG (Existence, Relatedness and Growth) theory by expanding the hierarchy of needs of Maslow. In the ERG theory, physiological and safety are the lower order needs and placed in the existence category. The relatedness category contains the love and external esteem needs. The growth category includes the higher order needs -self actualization and self esteem. Alderfer also proposed a regression theory to go along with the ERG theory. He said that when needs in a higher category are not met then individuals redouble the efforts invested in a lower category need.

Some studies showed that there is an overlap in the middle levels of the hierarchy of needs. Alderfer addressed this issue and reduced the number of levels. However, ERG needs can still be mapped to those of Maslow's theory as seen in Figure 4.1.

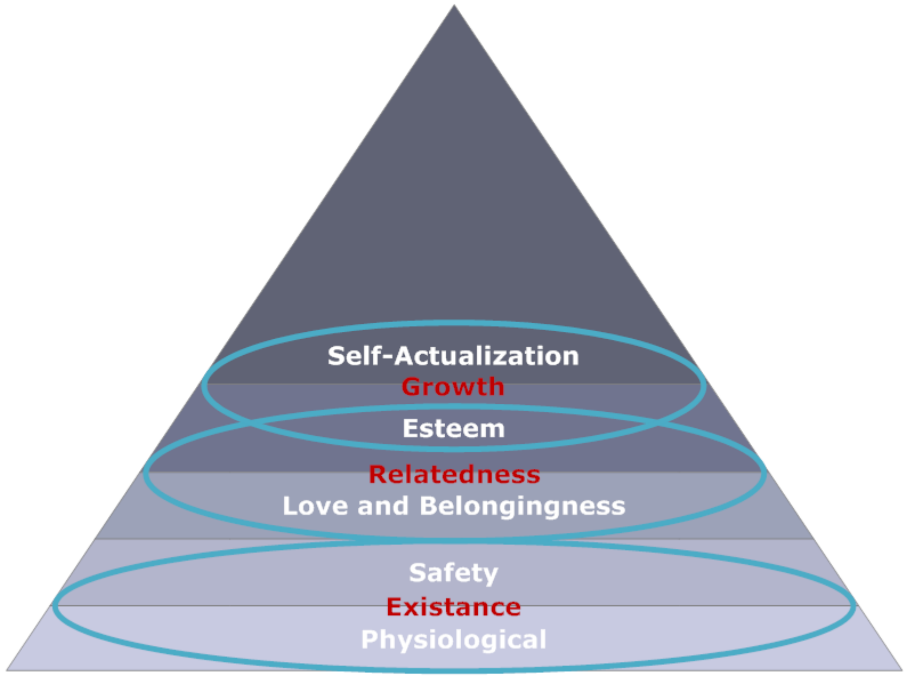

Figure 4.1: Hierarchy of Needs [Adapted from: Maslow [155] and Alderfer [7]]

ERG theory differs from the hierarchy of needs in the following three ways:

- The ERG theory allows for different levels of needs to be pursued simultaneously,
- The ERG theory allows the order of the needs be different for different people, and
- The ERG theory acknowledges that if a higher level need remains unfulfilled, the person may regress to lower level needs that appear easier to satisfy. This is known as the frustration-regression principle.

The hierarchy of needs is shown in Table 4.1. In this table, Alderfer's hieararchy is mapped to Maslow's hierarchy.

Another well known needs theory is called three needs theory. This theory is based on the ideas of Henry Murray [164] who identified the core psychological needs as achievement, affiliation and power. Accordingly, McClelland [159] put forward the three needs theory.

Table 4.1: Hierarchy of Needs [Adapted from: Maslow [155] and Alderfer [7]]

Alderfer's Hierarchy	**Maslow's Hierarchy**	**Associated Needs**
Existence	Physiological	Survival Air Water Food Excretion Warmth Sleep Sex
	Safety	Security Health and Well-Being Stability Religion
Relatedness	Love and Belongingness	Affectionate Relationships Involvement with Family Involvement with Friends Involvement with Others
	External Esteem	Being Needed Recognition Dignity

Continued on next page

Table 4.1 – continued from previous page

Alderfer's Hierarchy	Maslow's Hierarchy	Associated Needs
		Dominance
Growth	Self Esteem	Confidence Independence Achievement Mastery
	Self Actualisation	Know and Understand Fulfill Potentials Transcendence Wholeness

The three needs theory envisages that human-being have dominant needs for three things which influence their behaviour and these needs are:

- Need for Achievement,
- Need for Power, and
- Need for Affiliation.

According to the three needs theory, people with a high need for achievement seek to excel and tend to avoid the situations which include risks. Generally, achievement-motivated individuals avoid the low-risk situations which yield easy success not a genuine achievement. The need for affiliation is similar to the Maslow's belongingness need. People with a high need for affiliation need to have social relations and be accepted by other people. According to the theory, the need for power can be either personal or institutional. People who need personal power want to direct other people; while people who need institutional power want to organise the efforts of the others.

According to his study, subjective importance of each need varies from individual to individual. The importance of each need depends also on an individual's cultural background. In other words, an individual's specific needs are acquired over time and are shaped by one's life experiences. Therefore, McClelland's theory sometimes is referred to as the learned needs theory. It must be noted that Murray's contribution is rarely acknowledged in contemporary academic literature. It can be put forward that McClelland's theory is somehow weaker than the other theories; since, McClelland's theory is based only on Murray's ideas.

According to the cognitive theories, a motive sensitizes a person to information relating to that motive. As an example, a hungry person will perceive food stimuli more than any other stimuli. These stimuli motivate people to satisfy them. The pioneering cognitive motivation theory is called cognitive dissonance theory which is developed by Festinger [85]. In the cognitive dissonance theory, if a person experiences conflict or discrepancy between his beliefs and/or actions, then the person will act to resolve those conflicts and discrepancies. As a result, this will lead to a change in thought pattern which in turn leads to a change in the behaviour.

The expectancy theory is one another cognitive theory for motivation [228]. According to the theory, there are three factors called expectancy, instrumentality, and valance. Expectancy is explained as the perceived probability of success. Instrumentality stands for a connection of success and reward. Valance is the value of obtaining goal.

When values of these three factors multiplied, the value of the motivation is obtained. A low value in one of these will result in a low motivation. With this respect, all three must be present to motivate people. From this perspective, all three variables must be high in order to motivate a person.

The last distinctive theory of motivation is the goal-setting theory. It is based on the notion that the individuals sometimes have a drive to reach a clearly defined end state [149]. Often, this end state is a reward. A goal's efficiency is affected by three features; proximity, difficulty and specificity. An ideal goal should present a situation where the time between the initiation of a behaviour and the end state is close in time. A goal should be moderate, not too hard or too easy to complete. In both cases, most people are not optimally motivated, as many want a challenge (which assumes some kind of insecurity of success). At the same time, people want to feel that there is a substantial probability that they will succeed. The specificity concerns the description of the goal. The goal should be objectively defined and intelligible for the individual.

As it can be seen, there are many motivation theories and each of them explains different aspects of the motivation. Actually, in the theories of motivation there is no exact solution to understand the exact motives that drive intelligent-beings. For the purposes of the present study, the theories of needs are adopted. Specifically, under the light of the findings underlined by Maslow, Alderfer's ERG is adopted. It is because of the fact that the theories of needs are well defined and well known approaches to explain motivation.

Even though the ideas of Maslow are adopted in the proposed approach, it is also assumed that there is no difference between human and animal motivation. In other words, the proposed approach also accepts Freud's assertion which states that there is no difference between human and animal motivation. From this point of view, for all intelligent entities, it is proposed that the emergence of the intelligent behaviour is the result of motives like needs. With this respect, according to the proposed approach, driven by motives all intelligent beings act intentionally.

To achieve the simulation of intelligence, Alderfer's ERG theory is adopted; since, it is based on the most commonly approved approach called as the hierarchy of needs of Maslow. Besides, ERG expands the hierarchy of needs and resolves the overlapping in the middle levels of the hierarchy. Furthermore, in the current approach, it is proposed that the order of the needs can be slightly different for different beings as suggested by Alderfer. According to the proposed approach, this provides personality to intelligent beings.

It must be noted that the behaviours of all intelligent entities are different from each other. It may be the result of the fact that all human's needs are not exactly in the same order as listed by Maslow. As proposed by Alderfer, the needs of intelligent beings can be different from the others.

Moreover, different animal species' behaviours are also totally different from other species. These behavioural differences between different species are the result of the differences between the motives and physical capabilities. It is quite obvious that the motives of the animals are much more primitive than the human. In turn these primitive motives yield less complicated intelligent behaviours.

Moreover, it must be stressed that intelligent behaviours are limited with the physical capabilities. As an example, cats and dogs can be compared. It is reported that cats are capable to open closed doors while dogs are unable [222]. It is because of the fact that jumping capabilities of the cats are superior than the dogs and the cat paws are more flexible. Even if cats can open the closed doors, this does not mean that they are more intelligent then dogs or vice versa.

Moreover, human-beings can open both closed doors and locked doors. But cats cannot open a locked door. This also should not mean that human-beings are more intelligent then cats. This only shows that there is a difference between the physical capabilities of these species. In fact, Gould and Gould [101] asserted that there is a possibility for the cats for being more intelligent than human-beings!

4.2.4 Intentionality

According to this study, being driven by their motives, the intelligent beings act intentionally to satisfy those motives. In particular, the intentional notion provides a way to explain, understand and mimic intelligent behaviour. With this respect, being intentional is one of the most important attributes of intelligent beings. Moreover, it provides a good theoretical infrastructure for agency.

As stated before, the proposed approach attempts to combine the intentional notion and the theories of needs. In this manner, the emergence of intelligent behaviour is tried to be explained as a result of the motives (i.e. needs). Therefore, it is proposed that being driven by needs intelligent entities act intentionally to satisfy their needs.

In the new approach to illustrate the intentional notion Belief, Desire, Intention approach is adopted. As explained previously, the beliefs are the information that an agent has about itself and its environment. The desires are the possible alternatives that can be chosen by an agent. The intentions are the choices of an agent. In other words, intentions are the plans to which an agent has committed. The desires of an agent are represented in the form of plans. Plans are sequences of actions that an agent can perform to achieve its intentions.

According to the given information, the proposed approach works as follows: Firstly, intelligent entities observe their internal state and the environment; the perceived inputs are the beliefs of the entities. These beliefs are also at the same time the cause according to the causality assumption. Then intelligent entities check the relevancy of these beliefs with their needs. If the beliefs are related with their needs; then they become motivated to satisfy the relevant need. The goal of intelligent beings is then to satisfy the corresponding need.

To pursue their goals, intelligent beings try to determine their desires. For this purpose, they develop several plan alternatives. Afterwards, based on some criteria (which are explained in the following subsection), they select one of the plan alternatives as their intention. Then they perform actions corresponding to the intention in the environment to satisfy their need.

According to this viewpoint, the perceived input is the cause while the actions taken are the effect. In this manner, the proposed approach works under the causality assumption. Furthermore, by the proposed approach, the intentional notion and the theories of motivation are combined.

As an example, the behaviour of the man who closed the door to keep his cat away from the parlour can be examined. Here how the proposed approach works:

Once again consider Jack who is going to go out. Let us assume that there are flowerpots in the parlour and he sees the cat that is playing around them. If the cat is going to be left in the parlour, the cat will mess up the flowerpots. Therefore, the room will lose its hygiene. Living in a hygienic place is related with existence needs. Therefore, before he goes out, he must keep the cat away from the parlour. Keeping the cat away from the parlour becomes the goal of Jack. Then he starts developing plan alternatives.

Let us assume that he develops three different plans which meet his goal:

Plan 1. Close the door of the parlour and leave the cat in the corridor.

Plan 2. Tie the cat in the corridor.

Plan 3. Take the cat to the outside.

Then Jack tries to make decision on these plan alternatives. In the decision making process, he selects one of the plan alternatives as his intention. Let us assume that for the given example, Jack decided to close the door of the parlour and leave the cat in the corridor. This means that the intention of Jack is to close the door of the parlour and leave the cat in the corridor. After determining the intention, Jack executes its intention to satisfy its need.

4.2.5 Flexibility

Jennings et. al. [119] explains the term flexibility as the capability in performing flexible actions. They further elaborate this attribute by putting forward three components of it: responsive, pro-active and social. The term responsive refers to those entities that can understand their environment and respond to the changes that occur in their environment. The term pro-active refers to those entities that can perform actions and take initiative to achieve their objectives. In their definition, the term social implies beings that are able to interact with the other entities and also help the others in their activities.

This study follows Jennings et. al. who stated that one of the key attributes of intelligent entities is flexibility. However, in the present study, flexibility and social ability are considered as separate attributes. Therefore, in this study the term flexibility is used to refer to those entities which are responsive and pro-active. However, the social ability is considered as a separate attribute which is elaborated on in the following subsection.

According to this point of view, intelligent beings are capable of selecting different actions under different conditions. Even under the same conditions, intelligent beings have such flexibility that they select different courses of actions in accordance to the changes in their environment. In other words, the behaviours of intelligent beings are so unpredictable that they can choose to execute different plan alternatives. Therefore, intelligent decision making to choose among alternative plans is unpredictable even if the causes are known.

In the present approach, each decision is made to satisfy a particular need. As stated before, intelligent behaviour is considered as in accordance with the causality. Therefore, in the present approach the needs are proposed as nexi which provide the means to select among alternatives. Besides, it is proposed that intelligent beings select the plan alternative which satisfy their needs best.

With this respect, in this approach each plan alternative corresponds to a satisfaction degree related with a particular need. Here, the degree of satisfaction is a metric which provides the means to select among different plan alternatives. Each action taken does not guarantee full satisfaction of the needs. In some cases, intelligent-beings can satisfy their needs partially. The degree of satisfaction can be considered as a value between 0 and 1, while 0 signifies full dissatisfaction, 1 signifies full satisfaction.

Here the question is how to provide the required flexibility to the agent by using these metrics. Once again consider the example given in the previous subsection. Jack has three plan alternatives and assume that the satisfaction degrees of these plan alternatives are as follows:

Plan 1. Satisfaction Degree = 0.9

Plan 2. Satisfaction Degree = 0.7

Plan 3. Satisfaction Degree = 0.6

If each satisfaction degree is defined as a fixed value lying between 0 and 1 like above; there is no need in defining every alternative. Because given a fixed value, Jack would always select the same alternative, whose Plan 1 has the highest satisfaction degree. Therefore, if the satisfaction degrees are defined as fixed values; it means that the agent has no choice. In other words, the second and the third alternatives are not accessible. It is because of the fact that an agent always select the alternative that satisfies its needs best. As stated before, according to the proposal the agents choose the alternative plan which satisfies their needs best. From this point of view, by assigning fixed satisfaction degrees, the required flexibility cannot be provided.

To overcome this obstacle, probabilistic causality can be employed. The probabilities corresponding to each plan alternative can be defined. For the given example, assume that the following probabilities to each alternative are assigned:

Plan 1. $P_1 = 0.4$

Plan 2. $P_2 = 0.3$

Plan 3. $P_3 = 0.3$

This probabilistic causality seems more realistic; therefore, several existing cognitive architectures adopt this approach. But this time, defining probabilities is a very difficult issue. To overcome this problem some researchers adopt probability matching techniques. Even if these techniques are useful, still there are some other problems.

One of the problems is that when probabilities are explicitly defined, the behaviour of the agent is no longer unpredictable. For the given example, if a simulation is run for hundred times; then with high probability the agent would select the first alternative for 40 times. Likewise, it is going to be observed that the agent would select the second alternative for 30 times and the third alternative for 30 times. This approach is quite predictable.

In some cases, researchers utilize several different probability calculations and probability matching techniques to obtain unpredictable behaviour. Such approaches make behaviour quite unpredictable for the observers. But if the approach or calculations employed is known the behaviours of the agent can still be predicted. Therefore, this probabilistic approach does not provide the required flexibility.

Another problem is related with Agre's proposal. Agre stated that the most of the everyday activity is routine [5]. This means that intelligent-beings tend to choose the same plan alternative under the same conditions. In the frame of these references, to achieve the required flexibility while supporting routine activities, unpredictability must be maintained.

In the proposed approach, the probabilities cannot be utilized to define the satisfaction degrees. As it is expressed, in the present approach, needs are proposed as nexi. It is proposed to use the nexi as the evaluation metric which is the satisfaction degrees. Actually, satisfaction cannot be explained in terms of probabilities.

Satisfaction is not a probabilistic issue. It can be better explained in terms of a value with a certain mean value. For example, eating certain foods results in similar satisfactions. A man can compare his satisfaction obtained from eating two different foods. If it is tried to compare these satisfactions, these satisfactions can be evaluated on a scale. Therefore, in the proposed approach, a satisfaction degree is considered as a real value between 0 and 1, while 0 signifies full dissatisfaction, 1 signifies full satisfaction.

In this frame of reference, normal probability distribution function is suitable for explaining satisfaction phenomenon. The normal distribution is used as a first approximation to describe random variables. These random variables tend to cluster around a single mean value (like satisfaction which is explained above). The graph of the associated probability density function is known as Gaussian function [121].

Therefore, to provide a metric measure in the selection process, in this study random causality is proposed. The term randomness have several meanings. It has meanings like; having no definite aim or purpose, not sent or guided in a particular direction and done, made, occuring, etc... without conscious choice [72]. However, in this study randomness implies a lack of predictability. From this point of view, it can be said that in some sense the flexibility in human intelligence is result of randomness.

According to the proposal satisfaction degrees are introduced as normally distributed random numbers between 0 and 1 with certain mean and variance values. With this respect, each plan alternative corresponds to a satisfaction degree which is a normally distributed random number between 0 and 1 with certain mean and variance values. If each satisfaction degree is denoted by "ς", each mean value is denoted by "μ" and each variance value is denoted by "σ^2", by using previous example the approach can be represented as follows:

Plan 1. ς_1 = normally distributed random number with mean μ_1 and variance σ_1^2

Plan 2. ς_2 = normally distributed random number with mean μ_2 and variance σ_2^2

Plan 3. ς_3 = normally distributed random number with mean μ_3 and variance σ_3^2

According to this proposal, when an agent tries to choose among alternatives, the agent initially generates random numbers between 0 and 1 in accordance with the mean and variance value of the corresponding alternative. In this manner, the approach aims to provide flexibility in such a way that the behaviours of the agents cannot be predicted even if we know the mean and variance values of the satisfaction.

For the given example, assume that mean and variance values of each plan alternative are as follows:

Plan 1. ς_1 = normally distributed random number with mean $\mu_1 = 0.70$ and variance $\sigma_1^2 = 0.20$

Plan 2. ς_2 = normally distributed random number with mean $\mu_2 = 0.70$ and variance $\sigma_2^2 = 0.20$

Plan 3. ς_3 = normally distributed random number with mean $\mu_3 = 0.60$ and variance $\sigma_3^2 = 0.30$

By using these mean values, normally distributed random numbers can be generated as the satisfaction degrees (ς) of the plan alternatives. Assume that satisfaction degrees are generated by using a random number generator as follows:

Plan 1. $\varsigma_1 = 0.653$

Plan 2. $\varsigma_2 = 0.785$

Plan 3. $\varsigma_3 = 0.575$

According to these satisfaction degrees, Jack would select the alternative that best satisfies his need. The highest satisfaction degree implies that the corresponding plan alternative meets the need best. For the given example, Jack would select the second alternative; since, Plan 2 has the highest satisfaction degree.

Apart from this, the proposed approach can be adopted to support routine activities. By this mechanism, the variance values for each plan alternative can be defined separately. The lower variance along with a higher mean value would support the routine activities.

For the above example, a routine activity can be defined. Assume that Plan 1, keeping the cat outside the parlour, is wanted to be a routine activity. To simulate this routine activity, as an example the mean and variance values can be changed as follow:

Plan 1. $\varsigma_1 =$ normally distributed random number with mean $\mu_1 = 0.80$ and variance $\sigma_1^2 = 0,05$

Plan 2. $\varsigma_2 =$ normally distributed random number with mean $\mu_2 = 0.60$ and variance $\sigma_2^2 = 0.10$

Plan 3. $\varsigma_3 =$ normally distributed random number with mean $\mu_3 = 0.50$ and variance $\sigma_3^2 = 0.10$

By using these mean values, assume that generated satisfaction degrees (ς) are as follows:

Plan 1. $\varsigma_1 = 0.817$

Plan 2. $\varsigma_2 = 0.597$

Plan 3. $\varsigma_3 = 0.513$

As it can be seen in the example, the mean values of the second and third plan alternatives are reduced while increasing the mean value of the first plan alternative. At the same time, the variance value of the first plan alternative is reduced which means that there in not much variation in the satisfaction of this activity. The variance values of the other plans are also reduced. In this manner, as seen in the generated satisfaction degrees, the first plan becomes a routine activity; since, the man would select the plan alternative with the highest satisfaction. At the same time, there is still room for unpredictability.

When one of the alternatives is selected, it becomes the intention of the agent. After determining the intention, Jack starts executing the plan. In other words, he is in the intention to "close the door". After executing it, closing the door becomes the effect.

As it can be seen in the example intelligent behaviour is generated in accordance with causality. Here, this kind of causality is denominated random causality. Instead of the deterministic causality or the probabilistic causality, in the proposed approach random causality is adopted.

In the proposed approach decision-making is seen as a process in which intelligent entities produce effects due to some causes while the needs are the nexi. These causes are the input data related with the observation of either the internal state or the external world. In this process, the needs which provide a metric to measure different alternatives are put forward as nexi. When executed the actions lead to the effects. According to this approach this whole cycle is observed as intelligent behaviour.

As it can be seen in the example, by this approach the theories of need and the intentional stance are brought together. Here, the needs of intelligent entities motivate them to produce intelligent behaviour. With this respect, the needs drive the emergence of intelligent behaviour.

Even though, behaviours of animals and human-beings differ distinctively due to having different physical capabilities, most of their behaviour can be explained by employing the theories of needs, the intentional notion and random causality. As it can be seen in the given example, human-beings do not make decision in accordance with rationality at least not all the times. But they act intentionally to satisfy their motives. While trying to satisfy them, each action taken is not supposed to be rational like the behaviours of a man who worships so-called false gods! Along with the theories of the needs, random causality opens a channel to simulate this idea.

Another issue related with the flexibility attribute that is adopted in the proposed approach is conditional pro-attitudes. Intelligent beings also have beliefs, obligations, likes and dislikes that are related with certain conditions. Such pro-attitudes have impact over the satisfaction degrees of plan alternatives. Therefore, in the present approach conditional pro-attitudes which are associated with certain conditions are adopted. These conditional pro-attitudes have effect on the satisfaction degrees. Hence, the conditional pro-attitudes have influence on the selection of the relevant plan alternatives. To illustrate this influence, in our approach certain values to conditional pro-attitudes are assigned. In the present study, these values are called impact factors (ψ).

To illustrate this idea, once again consider Jack who has a goal to keep his cat away from the parlour. He may believe that tying the cat is an inhumane behaviour. According to this belief his satisfaction obtained from the second plan alternative should be reduced. In the proposed approach, the reduction is performed by utilizing an impact factor. In the end, Jack would tend to select the other available options.

In this approach, these impact factors can be a real number between -1 and 1, while negative values signify negative impact, positive values signify positive impact. In particular, a pro-attitude which has a negative impact factor reduces the mean value of the satisfaction degree; a positive impact factor increases the mean value of the satisfaction degree.

For the given example, let us assume that Jack believes that tying the cat is an inhumane behaviour. Therefore, the satisfaction obtained from the second plan alternative should be reduced. Consider that this conditional pro-attitude "Tying a cat is an inhumane behaviour." has impact factor ($\psi = -0.50$).

In the proposed approach, by using the given impact factor and the mean value of the satisfaction of the second plan, the ameliorated mean value of the satisfaction degree for the second alternative is calculated as follows: The mean value ($\mu_2 = 0.60$) is multiplied with the impact factor ($\psi = -0.50$). Then, this value is subtracted from the mean value; since, the impact is negative. Then the ameliorated mean value can be obtained as ($\tilde{\mu_2} = 0.30$) for the second plan alternative.

If there are other conditional pro-attitudes related with the other plans, the ameliorated mean values for these plans should also be calculated. Then these ameliorated mean values of the satisfaction degrees are assigned to the corresponding plans as their new mean values of the satisfaction degrees. Then, by using the mean values of the satisfaction degrees and the variance values, normally distributed random numbers between 0 and 1 are generated. By using these generated satisfaction degrees, the agent makes a decision.

4.2.6 Social Ability

Another important attribute of intelligent beings is that they have social ability. They can communicate with each other and they can collaborate on several different activities. Jennings et. al. [119] explained social ability as an attribute of beings that are able to interact with other entities and also help others in their activities. Even though they suggest this attribute as part of flexibility, in this study social ability is considered as a separate attribute.

To imitate the social aspect of the intelligence, it is assumed that when intelligent beings have common motives (i.e. needs), they tend to cooperate. Intelligent entities are well-adjusted if they do not have contradicting pro-attitudes. In other words, intelligent entities are so harmonious with the others that they avoid being in conflict with others while trying to satisfy their common needs. Therefore, they would rather avoid conflicts unless they have pro-attitudes which dictate otherwise. If they do not have contradicting pro-attitudes, intelligent beings would rather collaborate.

To understand the proposed approach better, consider a couple, Jack and Jane who are hungry. Initially, consider the plan alternatives of Jack. Assume that he has two different plan alternatives:

- Going out for dinner.
- Asking Jane if she can cook something.

After comparing satisfaction degrees on these plan alternatives, assume that Jack decided to ask Jane if she can cook.

At the same time, Jane should also has some plan alternatives. Assume that she has the same plan alternatives. If she has chosen the same plan, then there will be no problem. After communicating with each other, Jane is going to cook food for them. But if she has selected the other plan, then they should resolve the conflict.

Initially, Jack offers his desire, asking if she can cook some food. According to the proposed approach, Jane would accept the offer because she is well-adjusted. But assume that she has a conditional pro-attitude which is "Jane is tired". In such a case, she has a contradicting conditional pro-attitude. Therefore, she would not accept the offer and she would propose her desire and she would ask if they can go out by providing the relevant conditional pro-attitude.

This time, Jack will once again think about the option proposed by her. If there is no contradicting conditional pro-attitude, he would accept her offer and decide to go out. Assume further that he has an additional pro-attitude which says that "Jack does not have enough money to go out". Then, once again he will choose the first plan. Along with his conditional pro-attitude, he will once again suggest Jane to cook at home.

In this case, Jane receives the additional conditional pro-attitude. In this case, she has two contradicting pro-attitudes which also contradict each other. This time, she would try to compare these contradicting beliefs whichever is more influential, she will make decision in guide of these pro-attitudes. For instance, in this case, assume that the conditional pro-attitude that dictates “Jack does not have enough money to go out” has stronger influence. Therefore, the satisfaction of going out is going to be reduced and she will decide to cook at home. This negotiation can continue until a common plan is found to satisfy the common need.

As it can be seen in the given example, according to the proposed approach, the agents try to collaborate while trying to meet their common needs. This type of cooperation is similar to joint intentions approach. In the present approach, both agents are well-adjusted and try to find a common plan. But as it can be seen in the example conditional pro-attitudes have influence on the negotiation process. In case of a contradiction, the issue becomes complicated but to satisfy the common need, they tend to cooperate.

For conflict resolution, the conditional pro-attitudes which have certain impact factors are employed. The pro-attitude with higher impact factor has a stronger effect on selecting or not selecting the corresponding plan alternative.

It must be noted that certain pro-attitudes can have more influence than the others. For example, assume that Jack does not like his manager. Since he doesn’t like his superior; he would not want to cooperate. But whenever his manager requests something, he would choose to do it. It is because of the fact that he has an obligation which dictates he should do whatever his superiors say. Therefore, it can be stated that obligations have more influence than likes and dislikes.

4.2.7 Learning

Another important attribute of intelligent beings is that they can learn. As stated before, here the main objective is to achieve a general framework for simulating the intelligence of intelligent beings. When learning of them is observed, it can easily be deducted that intelligent beings are capable of supervised and unsupervised learning.

As an example, once again cats that can open a closed door can be considered. Cats learn how to open such doors from their masters'. Cats observe their masters' putting their hands on the handle and pushing the handle downward while opening the closed doors. After this observation, cats learn it and they are physically capable of putting it into practice. They put it into practice slightly differently due to the physical differences between their masters' and themselves. But they learn to open a closed door by pushing the door handle downwards. This type of learning is called unsupervised learning.

Another learning type is supervised learning which can be observed in intelligent beings. As an example, consider a cat taken from streets to live at home with some people. When a cat is brought to a house; a toilet for it at home is certainly required. Cats in the streets use soil as a toilet; therefore, something similar is needed. Most of the people use a box shaped toilets which are specifically provided for the cats. Then sand or silica sand is put into the box. The silica sand is not exactly the same as the natural sand but cats can use it for the same purpose. Teaching a cat how to use silica sand is quite easy. If the paws of the cat are rubbed on the silica sand for a few seconds, then the cat learns that it is a toilet. Afterwards, the cat starts using it. This type of learning is called supervised learning.

It must be stated that not all the animals are capable of both types of learning. But some of them at some certain level of intelligence are capable of both. In this context, it is proposed that a framework for simulating general intelligence should enable simulating both learning types.

In this study, it is proposed that an agent should be capable of both learning types. In the proposed approach, the only concern is the learning that takes place among agents by observing others (i.e. observational learning) and receiving plans from the others. For this purpose, social learning theory which focuses on learning in a social context is adopted [21]. The principles of social learning theory can be summarised as follows [174]:

- Agents can learn by observing the behaviours of others and the outcomes of those behaviours.
- Learning may or may not result in a behaviour change of an agent.
- Expected reinforcements or punishments can have effect on the behaviours of an agent.

Additionally, Bandura [21] suggests that the environment reinforces social learning. He states that this can happen in several different ways such as:

- A group of agents with strong likelihood to an agent can reinforce learning from them. For instance, a group of planning agents who use a hybrid planning approach can reinforce the other planning agents to learn the same approach.
- An individual third agent which have influence on an agent can reinforce learning from the other agents. As an example, a planning manager agent can reinforce one of the planning agents to learn a hybrid planning approach from the other agents.
- The expectation of satisfaction from a behaviour that is performed by the other agents can reinforce an agent to learn. The agent can observe that the other agents create plans faster than itself due to the use of the hybrid planning

approach. In turn, the agent would be reinforced to learn to use the same approach to create plans faster.

Social learning theory is adopted to simulate the intelligence of animals including human-beings. In particular, it is proposed that an agent should be able to learn plans by observing the other agents in unsupervised manner and get plans directly from the others to support supervised learning.

In this approach, reinforcement learning is adopted to simulate social learning. According to reinforcement learning, the agents learn a policy of how to act given an observation of the world. The policy maps the states of the world to the actions that the agent ought to take in those states [220].

An agent can learn either in a supervised or an unsupervised manner. The reinforcement in the present approach is provided by the predicting the satisfaction that can be obtained by using the plan to be learned. If an agent considers that the plan is sufficiently satisfactory then the agent starts learning. Otherwise, the agent does not learn the plan. Subsequently, the proposed supervised learning approach is explained, then the proposed unsupervised learning approach is elaborated.

In the supervised learning an agent receives a complete plan from another agent. Since reinforcement learning is adopted, the plans include the conditions and the actions. In addition, in this approach when providing a plan to another agent, an agent provides the associated need. In the proposed approach, reinforcement realised by the satisfaction.

To put in practice, a social context to an agent is needed to be established. For this purpose, in this approach every agent is grouped. For instance, if a real life like environment is simulated, the social context can be established as it is shown in Figure 4.2.

In the figure, from the inner circle to the outer circles, influence on the agent is reduced. This means that the agent can be reinforced more by its family, while it

Figure 4.2: An Agent in Social Context

cannot be reinforced that much by the other agents. As proposed by Bandura, the agents with influence and group of agents with similarities are introduced.

It must be noted that, from one agent to another the social context may vary. As an example, an agent might be more influenced by its family, while another agent might be more influenced by its friends which are in the agents with similarities category.

Consider two planning agents: One of them might be more influenced by the planning manager agent which is a member of agents with influence. The other agent might be more influenced by the other planning agents who are members of agents with similarities. Even though the influence of a planning manager agent varies, in both cases the planning manager agent can reinforce learning.

In the proposed approach, learning is associated with one of the needs of the agent. In particular, it is associated with knowing and understanding needs. In supervised learning, whenever an agent receives a plan from another, a goal “learn” is generated. In this situation, the first agent can be called the learner agent, while the latter is the teacher agent. If the learner agent is not pursuing any lower level need, then the learner agent starts learning in a supervised manner.

When an agent learns in a supervised manner, it first assigns an expected mean value of the satisfaction degree and an expected variance value to the plan that is being learned. These values signify the expectations of an agent. To do so the agent considers the teacher agent in a social context. To provide reinforcement in this approach, each level illustrated in Figure 4.2, corresponds to an expected mean value of the satisfaction degree $E(\mu)$ and an expected variance value ($E(\sigma^2)$). The default values in the proposed approach are as follows:

The Family of the Agent: $E(\mu) = 0.70$, $E(\sigma^2) = 0.05$

The Agents with Influence: $E(\mu) = 0.60$, $E(\sigma^2) = 0.05$

The Agents with Similarities: $E(\mu) = 0.50$, $E(\sigma^2) = 0.10$

The Other Agents: $E(\mu) = 0.40$, $E(\sigma^2) = 0.10$

As it can be seen, while moving from the inner circle to the outer circles, the expected mean values of the satisfaction degrees are decreasing and the expected variance values are increasing. The expected mean value represents the expected average satisfaction that is going to be obtained by executing the learned plan. The expected variance value signifies the expected variation in the satisfaction that is going to be obtained by executing the plan learned.

These mean values and variance values are relatively defined to distinguish the differences of the influence on an agent. Here, the purpose is to enable an agent to learn from the others. These values provide an initial reinforcement to an agent to encourage them to learn from the others.

As explained before, the social context of an agent may vary from one to another. Therefore, for each individual agent these mean and variance values may also vary relatively. As an example, if an agent is more influenced by an agent with influence than its family, the expected values should be changed accordingly.

When defining these values, two principles offered by Bandura are considered [20]. Firstly, Bandura proposes that an agent is more likely to adopt behaviour if it results in outcomes it values. The second principle of Bandura is that an agent is more likely to adopt a modelled behaviour if the model is similar to the agent and has admired status. By the proposed approach, these principles are adopted.

While establishing social context, it is simply assumed that the family has more similarity and has more admired status to an agent than the agents with influence, the agents with similarities and the other agents. Likewise, the agents with influence have more similarity and more admired status than the agents with similarities and the other agents. Finally, the agents with similarities have more similarity and more admired status than the other agents. Once again it must be noted that this social context may vary from one agent to another. However, for the purposes of this study, the proposed context is employed.

In supervised learning, the learner agent directly receives the plan and the associated need. Whenever the agent receives the plan, it assigns the expected mean value of the satisfaction degree $E(\mu)$ as the mean value of the satisfaction degree to that plan. Likewise, the agent assigns the expected variance value ($E(\sigma^2)$) as the variance value to that plan. While assigning these values, the agents take social context into account and these values are assigned according to the place of the teacher agent in the social context. For instance, in the proposed approach, if an agent learns a plan in a supervised manner from an agent with similarities, then, the satisfaction degree of that plan will be $E(\mu) = 0.50$ and the variance value of that plan will be $E(\sigma^2) = 0.10$.

Naturally, unsupervised learning is different from supervised learning. Unsupervised learning, in the proposed approach, is observational learning. like in supervised learning, learning occurs only if the goal is "learn". Whenever an agent sees another one starting an action, a goal "learn" is generated. If the learner agent is not pursuing any other goal related with a lower level need, then the learner agent starts learning by observing the other agent.

In unsupervised learning, social context helps an agent to determine the initial reinforcement (R_i). If the teacher agent is a member of the family of the agent, then the learner agent believes that the learning plan from that agent is highly satisfactory. In other words, the expected mean value of the satisfaction degree for plans that is learned from family is $E(\mu) = 0.70$, while variance is $E(\sigma^2) = 0.05$. If this initial mean value of the satisfaction degree is satisfactory enough for the agent, then it starts learning from that agent by observing it.

To determine a satisfactory plan in the proposed approach, there is a learning satisfaction limit (ς_l^2). The default limit is (ς_l^2) $= 0.40$ to encourage learning from the other agents. If the mean value of the satisfaction degree is greater than or equal to the learning satisfaction limit, the agent starts learning. This limit ensures that an agent tends to learn from the other agents; since, all of the proposed expected values are greater than or equal to the offered limit.

For supervised learning, as it can be seen above, all satisfaction degrees and mean values are greater than and equal to learning satisfaction limit. Therefore, an agent at least starts to learn whenever it sees another agent in action. When a learner agent observes other agents, it sees each action one by one. The learner agent checks if it knows the observed action. If the agent knows it, then the agent simply does not need to learn it again. Otherwise, the agent learns the action. Each action requires pre- and post-conditions while the action is associated to a need.

As stated before, while starting unsupervised learning, the agent uses initial mean values of the satisfaction degrees and variance values in accordance with the group of the teacher agent. For instance, assume that an agent starts to learn a plan from another agent which is a member of the agents with influence. Since, the expected mean value of the satisfaction degree for this group is above the learning satisfaction limit, if the plan is not known by the agent, it starts learning the plan. According to this approach, when learning the new plan, the agent is going to assign the following initial mean and variance values to that plan: $\mu = 0.60$, $\sigma^2 = 0.05$.

When the agent starts learning by observation, it observes each action taken separately. After each action is observed, the learner agent checks if it knows the observed action. If the observed action is known, it results in a decrease in the mean value of the satisfaction degree obtained from learning the corresponding plan. If the observed action is not known, it yields an increase in the satisfaction.

To realise this idea, "reward" (r) and "punish" (p) values are introduced. The reward is $r = 0.20$ and punish is $p = -0.20$ by default. By using these values, after receiving each action the learner agent recalculates the mean value of the satisfaction degree. If the action is not known, it is rewarded by multiplying $1 + r$ and $E(\mu)$. If the action is known, it is punished by multiplying $1 + p$ and $E(\mu)$. The outcome of the multiplication becomes the new expected mean value of the satisfaction degree.

These calculations continue until either the plan is completely learned or the expected mean value of the satisfaction degree becomes less than the learning satisfaction limit (ς_l). During the learning process, whenever a satisfaction degree mean value becomes less than the learning satisfaction limit, the agent stops learning. If the expected mean value of the satisfaction degree stays above the learning satisfaction limit until the whole plan is learned, then the last calculated expected mean value of the satisfaction degree is assigned to the learned plan as its corresponding mean value. At the same time, the expected variance value is assigned to the learned plan as the variance value according to the social context proposed.

This type of learning in the proposed approach is denominated as "S-Learning" which stands for social learning by satisfaction reinforcement.

4.2.8 Ability to Display Affect

The final issue related with the proposed approach is to simulate the affective aspect of the intelligence. The proposed approach is put forward to cover many aspects of intelligent behaviour; therefore, the agent should also be affective. It is because of the fact that intelligent beings are capable of experiencing and displaying emotions.

In the proposed approach, affect is considered to be post-cognitive by following Lazarus [141]. According to this point of view, an experience of emotions is based on a prior cognitive process. As stated by Brewin [38], in this process, the features are identified, examined and weighted for their contributions.

Maslow [155] also considers emotions as post-cognitive. While explaining the needs, he states that if the physiological needs are not satisfied, it results in different emotional states like irritation, pain and discomfort. Hereby, his approach is extended by stating that the satisfaction or the dissatisfaction of not only physiological needs but also every need results in feeling different emotions.

While explaining emotions, the approach of Wukmir is adopted. Wukmir [242] proposed that emotions are such a mechanism that they provide information on the degree of favourability of the perceived situation. If the situation seems to be favourable to the survival of an intelligent being, then the being experiences a positive emotion. A being experiences a negative emotion, when the situation seems to be unfavourable for survival of the being.

When the theories of needs are considered, it can be claimed that the survival of the beings depends on meeting their needs. From this point of view, Wukmir's approach and theories of needs can be combined. According to this proposal, every need in the hierarchy can be associated with two different emotions. While one of these emotions is positive, the other one is negative. Whenever a particular need is adequately satisfied, it results in the generation of a positive emotion; since, it is favourable for survival. Likewise, if a particular need is not sufficiently satisfied, it results in a negative emotion. It is because of the fact that when a need is not satisfied, it is not favourable for survival.

As an example, consider Jack who is hungry. Need for food is one of the existence needs of all living beings. If Jack can sufficiently meet his need for food, he is going to feel a positive emotion, content. If Jack cannot sufficiently satisfy this need, he is going to feel a negative emotion, anxiety.

In the proposed emotion model, the emotions are classified in two levels: basic emotions and non-basic emotions. The basic emotions are the most primitive or universal emotions like pain, panic and anxiety. For the purposes of this study, the other emotions are considered as non-basic emotions such as love and loneliness.

Accordingly, the lowest level which includes the existence needs corresponds to the basic emotions. For example, if an agent meets its security need, this will result in the generation of comfort feeling. As long as the agent meets its security need, the agent is going to feel comfortable. In the ERG approach, the higher level needs are the existence needs and the growth needs. These needs correspond to the non-basic emotions. As an example, if an agent meets its affectionate relationship need, it is going to feel love.

In the frame of these references, it is proposed that emotions emerge when the needs of intelligent beings are satisfied or dissatisfied. The satisfaction of the needs results in the positive emotions, while the dissatisfaction of the needs results in the negative emotions. According to this viewpoint, while the satisfaction of the lower level needs triggers the primitive emotions; the satisfaction of the higher level needs results in the non-basic emotions.

To put the proposed approach into practice, each need is associated with a positive and a negative emotion. The needs and associated emotions are shown in Table 4.2.

Table 4.2: Needs and Associated Emotions

Associated Negative Emotion	**Needs**	**Associated Positive Emotion**
Existence Needs and Basic Emotions		
Pain	Survival	Ease
Panic	Air	Relief
Anxiety	Water	Content
Anxiety	Food	Content
Irritation	Excretion	Calmness
Anger	Warmth	Delight
Anger	Sleep	Delight
Discontent	Sex	Pleasure
Discomfort	Security	Comfort
Despair	Health and Well-Being	Expectancy
Sadness	Stability	Elation
Fear	Religion	Assurance
Relatedness Needs and Social Emotions		
Loneliness	Affectionate Relationships	Love
Loneliness	Involvement with Family	Love
Envy	Involvement with Friends	Joy
Envy	Involvement with Others	Joy
Embarrassment	Being Needed	Respect

Continued on next page

Table 4.2 – continued from previous page

Associated Negative Emotion	**Needs**	**Associated Positive Emotion**
Embarrassment	Recognition	Respect
Shame	Dignity	Pride
Shame	Dominance	Pride
Growth Needs and Non-Social Emotions		
Prejudice	Confidence	Detachment
Prejudice	Independence	Detachment
Mystery	Achievement	Familiarity
Mystery	Mastery	Familiarity
Confusion	Know and Understand	Discovery
Confusion	Fulfil Potentials	Discovery
Incompleteness	Transcendence	Completion
Incompleteness	Wholeness	Completion

To establish this table, the needs are mapped the emotions. Ortony and Turner [175] studied on the basic emotions and established a list of basic emotions from the literature. Therefore, while mapping the basic emotions, the emotions listed as fundamental emotions by Ortony and Turner are adopted. While there is a corresponding positive emotion for most of the needs, for a few of the needs a corresponding negative emotion in the list of Ortony and Turner cannot be found. Likewise, while there is a corresponding negative emotion for most of the needs, for a few of the needs, a corresponding positive emotion cannot be found. In such conditions, the corresponding positive or negative emotion is derived from the present one.

Although the lists of the basic emotions vary from a researcher to another researcher, it is simply assumed that the emotions mapped to the existence needs are the basic emotions of intelligent beings. It is because of the fact that whenever an agent meets the existence needs, they feel these universal emotions independent from the other factors.

According to this point of view, the other emotions are considered as the non-basic emotions. Therefore, for the higher level of the needs, the non-basic emotions are mapped with the needs. When considering ERG, it can easily be observed that the relatedness needs are external needs, while the growth needs are internal needs. Pursuing the relatedness needs involves other beings, while pursuing the growth needs does not involve interaction with the other beings.

Social emotions emerge wholly from the interpersonal concerns. In particular, social emotions occur only as a result of the encounters with other beings. However, non-social emotions emerge from the non-social events that do not involve interaction with other beings [223]. By considering these facts, the relatedness needs are mapped with the social emotions while the growth needs are mapped with the non-social emotions. It must be noted that some of the basic emotions can also be classified as social or non-social emotions. While mapping existence needs, the emotions are mapped regardless of this situation.

While mapping the social emotions with the relatedness needs, the list of the emotions established by Hareli and Parkinson is adopted [107]. In their study, Hareli and Parkinson list several social emotions proposed in the literature. Therefore, these emotions are mapped to the relatedness needs.

The non-social emotions are adopted from an artificial language called Lojban. Lojban is a constructed language designed to remove ambiguity from the human communication [64]. Therefore, the non-social emotions are derived from Lojban emotions and map them to the growth needs.

In Table 4.2, the needs employed and associated emotions are listed. The listed needs are the motives of human-beings. To establish this table, firstly the needs proposed by Maslow and Alderfer are gathered together and then several emotions proposed in the literature are mapped with those needs. As it can be seen in the table, several other emotions are not employed in the proposed approach. But one can introduce additional needs and associate more emotions with those needs.

In addition to these, Wukmir [242] states that emotions are expressed with a positive-negative scale and in variable magnitudes. For instance, one can say that "I feel quite calm", or "I feel calm" (in positive scale) or "I feel quite anxious", or "I feel anxious" (in negative scale). In this frame of reference, Wukmir proposes that all emotions consist of two components:

1. Quantitative Component: Indicates the magnitude of the emotion.
2. Qualitative Component: Indicates the description of the emotion which determines the positiveness or negativeness of the emotional sign.

These components can be mapped in accordance with the current proposal as it can be seen in Figure 4.3. In the figure, the emotions are categorised as positive and negative. While satisfaction results in a positive emotion, dissatisfaction yields a negative emotion. In addition, as illustrated in the figure, it is proposed that the degree of the satisfaction or the dissatisfaction determines the strength of the emotions.

According to this viewpoint, the emotions are categorised as negative and positive. Then, they are classified further according to their magnitude. For this purpose, two types of emotions are proposed as regular and strong emotions. To illustrate this approach, four different emotion limits are introduced. If a satisfaction obtained is above or below these limits, it results in the generation of four different types of emotions as shown below:

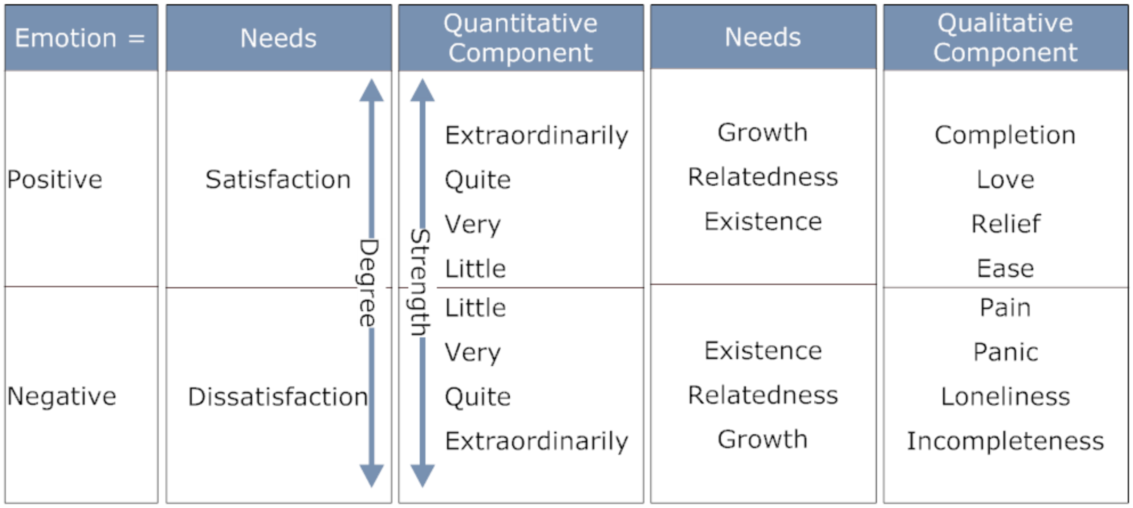

Figure 4.3: Components of Emotions [Adapted from: Wukmir [242]]

1. Strong Positive Emotion Limit: If the satisfaction degree is above this limit, it results in a strong positive emotion.

2. Positive Emotion Limit: If the satisfaction degree is above this limit, it results in a regular positive emotion.

3. Negative Emotion Limit: If the satisfaction degree is below this limit, it results in a regular negative emotion.

4. Strong Negative Emotion Limit: If the satisfaction degree is below this limit, it results in a strong negative emotion.

These emotions and emotion limits are visually represented in Figure 4.4.

To illustrate the idea, consider five agents who are looking for a house to meet their security need. Whenever an agent meets its security need, it feels comfort, while whenever an agent does not meet its security need, it feels discomfort. Assume that all of those agents found houses in different neighbourhoods. Further assume that their satisfactions obtained from the houses found are as shown in Table 4.3.

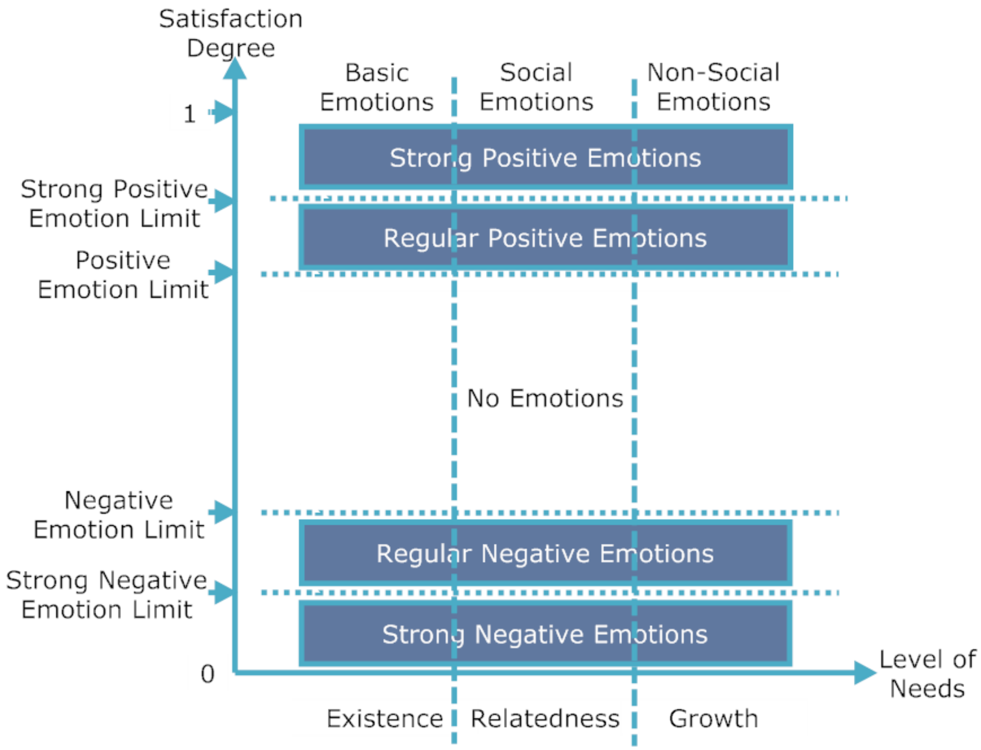

Figure 4.4: Emotions and Emotion Limits in the Proposed Approach

Table 4.3: Satisfaction Degrees of Agents

Agents	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Satisfaction Degrees	0,15	0,00	1,00	0,85	0,40

As it can be seen in the table, safety need of the first, fourth and fifth agent is partially satisfied. The second agent is fully unsatisfied, while the third agent is fully dissatisfied. For this example, assume that the emotion limits are as follows:

Strong Positive Emotion Limit = 0.90

Positive Emotion Limit = 0.80

Negative Emotion Limit = 0.20

Strong Negative Emotion Limit = 0.10

According to these limits, the emotions that are felt by these agents can be listed as follows:

Agent 1 feels regular discomfort; since, Negative Emotion Limit $> 0.15 >$ Strong Negative Emotion Limit

Agent 2 feels strong discomfort; since, Strong Negative Emotion Limit > 0.00

Agent 3 feels strong comfort; since, $1{,}00 >$ Strong Positive Emotion Limit

Agent 4 feels regular comfort; since, Strong Positive Emotion Limit $> 0.85 >$ Positive Emotion Limit

Agent 5 does not feel any emotions; since, Negative Emotion Limit $< 0.40 <$ Positive Emotion Limit

As stated before, the security need is related to the feeling of comfort and discomfort. Whenever an agent sufficiently meets its security need, it feels comfort. If an agent cannot sufficiently satisfy its security need, the agent feels discomfort. The strength of the emotion depends on the satisfaction degree.

Another issue related to the affective aspect of intelligence is the effect of the emotions on intelligent behaviour. In this study, it is proposed that the emotions have direct influence over the order of the needs. Therefore, except for the existence needs, in the proposed approach the order of the needs is not fixed. It is proposed that a strong emotion can change the order of the associated need. In particular, if a need is strongly satisfied or dissatisfied, it results in a strong emotion. While the strong positive emotion moves associated need downwards in the hierarchy, the strong negative emotion moves associated need upwards in the hierarchy.

If a need moves up in the hierarchy, it means that an agent is going to avoid the conditions associated with that need; since, the agent always pursues the lower level needs first. If a need moves down, it is more likely that an agent is going to pursue that need more frequently.

As an example, we can consider an agent who meets its achievement need. If its achievement need is strongly satisfied, the agent is going to feel familiarity more strongly. The more familiarity the agent feels; the more the associated need goes down. In this way, the agent is going to be more concerned about achievement. The agent is going to focus more on his works which satisfy its achievement need. In some extreme conditions, it can be observed that some people only think about their work and neglect their family and friends. It might be the result of this situation in which the achievement need became a lower level need than the other needs such as involvement with family and friends.

This proposal is also in accordance with the approach of Wukmir [242]. He states that the living organisms need to know if the conditions are useful and favourable for their survival. He proposes that the emotions are the mechanisms which provide means to know if a situation is favourable or not. He adds that by the help of the emotions, the living beings attempt to find favourable situations to survive which produce positive emotions. Likewise, they refrain from unfavourable states for survival which produce negative emotions.

The proposed approach provides the means to recreate this mechanism. Whenever an agent feels strong positive emotion, the associated need moves downward in the hierarchy. In this manner, an agent is more likely to pursue that need more frequently. Similarly, a strong negative emotion moves the relevant need upward, so that the agent can refrain from situations which are not favourable or unsatisfactory.

Moreover, in the proposed approach, the order of relatedness and growth needs can change. However, the order of existence needs can never change. It is because of the fact that these needs are significant for the survival of a being. For instance, a living organism cannot purse the affectionate relationship need before its need for food. Without eating food for too long, a living organism cannot survive.

In addition, the proposed approach provides the means to develop agents whose order of needs are different than those of the others. It can be observed that this approach provides personality to the agents. Besides, Alderfer [7] also proposes that the order of needs of beings can be different. The proposed emotion mechanism integrated with the hierarchy of needs illustrates this idea of Alderfer.

It must be noted that the proposed approach is put forward to simulate not only human but also animal intelligence. However, it is more likely that animals other than humans have less complicated needs like air, food, security, involvement, and so on. Besides, animals are capable of producing emotions as well. Since, they have lower level needs; they are more likely to feel the basic emotions like pleasure and pain.

4.3 Intelligence as a Whole

In this section, the key attributes of intelligent beings are summarised. The proposed approach attempts to cover all of these attributes while developing intelligent agents. In this subsection, these key attributes are brought together by redefining the term intelligence.

By utilizing the proposed key concepts, it is proposed that the term intelligence refers to an abstract notion to express the cognitive processes of autonomous, situated, flexible, and social entities which can display affect and learn while they perform activities intentionally that are motivated by their needs.

In this definition, eight attributes are regarded as the key attributes to explain intelligence. These are autonomy, situatedness, employing motives, intentionality, flexibility, social ability, learning and ability to display affect. To clarify the proposed definition of intelligence these terms can be explained as follows:

Autonomy: Reasoning and decision-making based on perception without direct or indirect control of other entities.

Situatedness: Existing in an environment in such a way that an entity can affect its environment by performing certain actions.

Employing Motives: Employing certain motives like needs to be motivated to generate goals.

Intentionality: Being intentional and acting intentionally.

Flexibility: Performing flexible actions in the boundaries of physical capabilities to react to the changes in the environment.

Social Ability: Being social in such a way that an entity can communicate and collaborate with other entities.

Learning: Ability to change attitudes and behaviours by being conscious of perceptions, decisions, and actions.

Ability to Display Affect: Being capable of experiencing and displaying emotions.

The first three attributes listed above are nearly the same as those proposed by Wooldridge and Jennings [240]. In addition, the learning attribute is adopted from Russell and Norvig [194].

The other attributes are put forward in accordance with the proposed approach. In particular, these attributes are derived from the core attributes provided in the literature. It must be noted that the proposed definition of intelligence is put forward to explain the general intelligence concept. With this respect, it tries to explain the human, animal and agent intelligence.

Chapter 5

ReCau: Reactive-Causal Architecture

According to the proposed approach, a general purpose architecture is proposed. This architecture is called Reactive-Causal Architecture (ReCau). Based on the foundations stated in the previous section, ReCau can be employed to simulate intelligent beings.

In this section of the study, the details of the Reactive-Causal Architecture are presented. Initially, the components and the mechanisms of the architecture are introduced. Afterwards, each layer of the architecture is explained in details. While explaining each layer, the components and the mechanisms are elaborated upon. In the last subsection, ReCau is compared with several existing architectures.

5.1 Overall Architecture

The proposed architecture is designed to be in accordance with the framework explained in the previous section. ReCau can be employed to develop agents which are highly autonomous, situated, flexible and social. Besides, ReCau agents employ motives and display affect. ReCau consists of three hierarchical layers: while the lowest layer is reactive, the highest layer is causal.

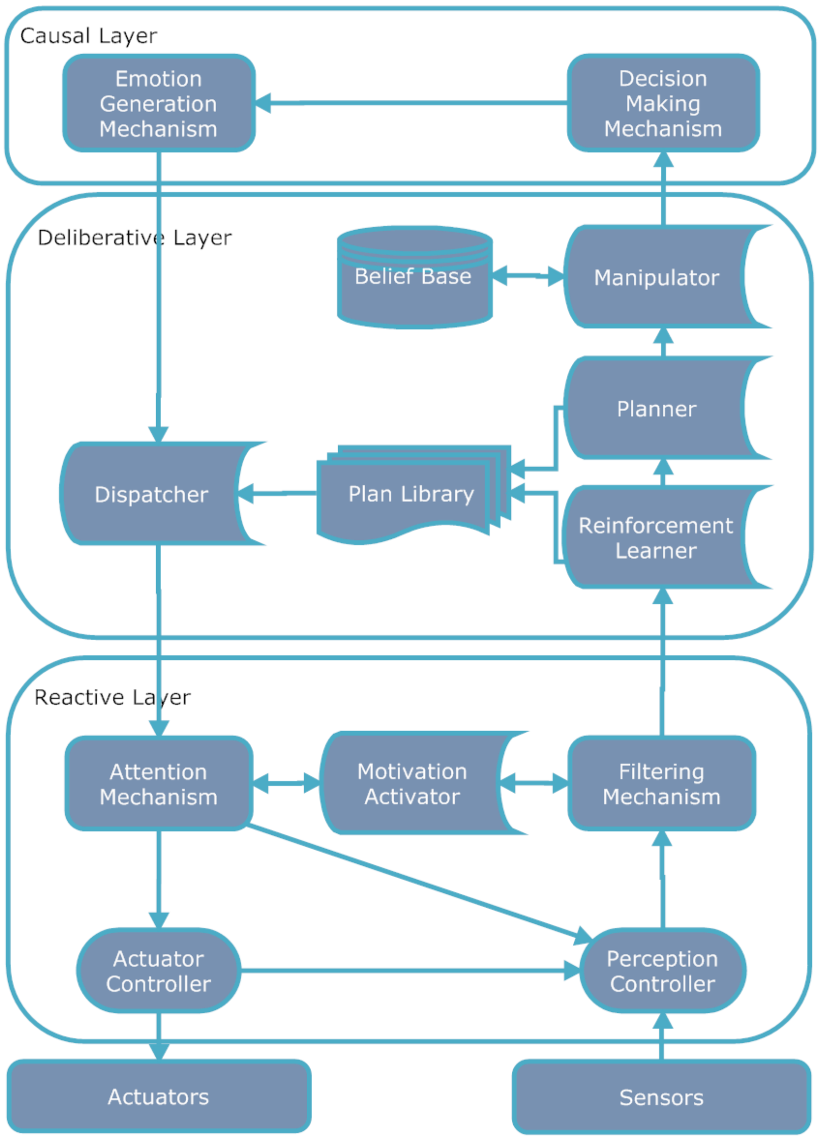

Figure 5.1: The Components of the Reactive-Causal Architecture

The reactive layer is in its classical form and it is meant to interface with the environment. It controls the perception and the action to monitor the internal and external conditions. The middle layer has deliberative capabilities such as action planning and task dispatching. Decision-making and emotion generation occurs in the causal layer. Each layer is capable of communicating with its neighbouring layer. The communication between the components and the layers are provided by simple message passing. The overall structure and the components of the architecture are shown in Figure 5.1.

As it can be seen in the figure, each layer of ReCau includes different mechanisms and components. The reactive layer includes a perception controller, an actuator controller, a filtering mechanism, a motivation activator, and an attention mechanism. The deliberative layer includes a reinforcement learner, a planner with a plan library, a manipulator with a belief base, and a dispatcher. The causal layer contains a decision-making mechanism and an emotion generation mechanism.

In general, a ReCau agent continuously observes the internal and external conditions by its reactive layer. The motivation activator holds the motives which are the needs of the agent. By the help of these needs, a ReCau agent generates goals to satisfy its needs. Regarding the goal of the agent, the attention mechanism directs the controlling components to focus on activities related with its needs. If the observations are not related with the needs of the agent, they are filtered out. If the observed conditions are related with the needs then they are sent to the deliberative layer.

In the deliberative layer, if it is required, reinforcement learning occurs based on the observations of the agent. According to the conditions observed, the agent learns plans and deploys them in the plan library. By the help of the plan library, the planner develops plan alternatives to achieve the goal. After developing these plans, if required the manipulator initially updates the pro-attitudes of the agent. These pro-attitudes are stored in the belief base. Along with the belief base, the manipulator determines the ameliorated mean values of the satisfaction degrees corresponding to particular plan alternatives. Besides, the manipulator enables the agent to resolve conflicts between the other agents.

In the causal layer, the agent makes decisions to select the most appropriate plan which satisfies its need and goal best. While making decisions, the needs of the agent provides the means to evaluate plan alternatives. The evaluation metric is the degree of satisfaction. These satisfaction degrees are obtained by using ameliorated mean values of satisfaction degrees and variance values. By comparing the degrees of satisfaction, the agent selects the plan alternative with the highest satisfaction degree. The selected plan is the intention of the agent. After determining the intention, if the need of the agent is sufficiently satisfied or not satisfied, the emotion generation mechanism generates an emotion.

Then the agent sends the intention and the emotion to the deliberative layer. According to the intention and the emotion, the deliberative layer informs the reactive layer on which actions are to be taken by its dispatcher. Finally, the agent starts executing its intention under the guidance of the reactive layer. If the intention is executed successfully and if an emotion is generated then the agent displays affect in accordance with the generated emotion. The detailed functions of these components and mechanisms are explained in the following subsections.

5.2 Reactive Layer

The ReCau architecture enables an agent to continuously observe internal and external conditions. In other words, the ReCau agents are able to monitor both external world and internal state continuously. This function is guided by the perception controller in the reactive layer. To perform its functions, ReCau requires additional external components called sensors. Traditionally, these components can be visual, auditory, tactile, proprioceptive, taste, or smell sensors.

In addition, another sensor is required to enable a ReCau agent to obsere internal state. This sensor can be called the meta-controlling sensor being inspired by Sloman [209]. The sensors that are going to employed must be selected in accordance with the purpose of the agent design. For example, if you intend to develop a humanoid robot to simulate human-like intelligence, then you must employ all of these sensors.

The perception controller in ReCau is meant to control these sensors. This component receives and processes sensory data continuously. Here, these data correspond to the conditions of both the external world and the internal state. After receiving the conditions, the perception controller sends data to the filtering mechanism.

The filtering mechanism is responsible for filtering out sensory data which is not related to the needs of the agent. Together with the motivation activator, the filtering mechanism puts condition-action rules into practical use. These condition-action rules include two sets of rules. The first set of rules is applied to eliminate data which are not related to the needs of the agent. As stated before, the perception controller continuously receives conditions and sends them to the filtering mechanism. If these conditions are not related to the needs of the agent, the filtering mechanism simply filters out those conditions. Otherwise, the motivation activator generates a goal to satisfy the corresponding need. Then the filtering mechanism sends the condition, the corresponding need and the goal to the deliberative layer. The need and the goal together are called the motive of the agent. The condition sent can also be called the pre-condition. The flow chart of the motive generation process is shown in Figure 5.2.

To realise this mechanism, each condition is related to a need. These needs and conditions are to be explicitly defined in the ReCau architecture. While defining these needs in ReCau, the needs listed in the previous section can be adopted. Even more needs can be defined in the implementation phase. While defining these needs,

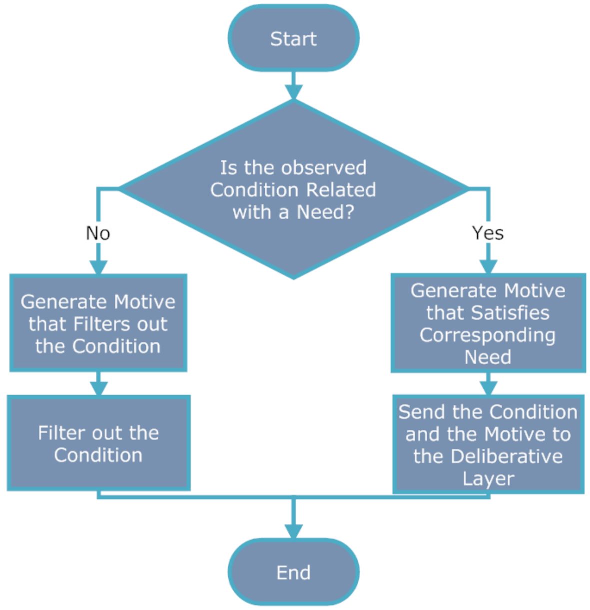

Figure 5.2: Flowchart of the Motive Generation Process

to put the proposed approach into practice, each need is put in a hierarchical order. This enables the ReCau agent to distinguish the order of different needs, even if they are in the same level of the hierarchy. By distinguishing the order of the needs, the ReCau agent is enabled to pursue the lowest level need first. The entity-relationship diagram of the hierarchy of needs is shown in Figure 5.3.

As an example, both need for food and need for security is at the same level of ERG which is existence. However, these needs must also be put in further hierarchical order in ReCau. For instance, it can be said that the need for food is a lower level need than the need for security. In this manner, ReCau is able to distinguish different needs even if they are at the same level of the hierarchy. Assume that a ReCau agent receives two conditions: the first condition is related to food, while the second condition is related to security. In such cases, the ReCau agent would try to satisfy its need for food first; since, the need for food is a lower level need than the security need.

Figure 5.3: The Entity-Relationship Diagram of the Hierarchy of Needs

The other function of the filtering mechanism is to notify the motivation activator on completion of actions. Whenever an action execution is completed, the actuator controller notifies the perception controller that the execution is completed. Then the perception controller verifies if the execution is completed. If the execution is completed, the perception controller warns the filtering mechanism. In accordance with this warning, the filtering mechanism informs the motivation actuator that the action has been executed.

The motivation activator generates the goals of an agent. The goals are held in a queue which includes the goals in a hierarchical order. The hierarchical order of the goals is determined in accordance with the level of the corresponding need. The goal related with the lowest level need takes the first order in the queue. The other goals, which are the goals related to higher order needs, are to be satisfied after the lowest level need is met. Whenever a need is satisfied, the corresponding goal is removed from the queue. Then the agent pursues the next goal in the queue.

While the agent is performing actions to satisfy its particular need, it can receive a condition related to a lower level need than the active need. In such cases, the agent puts the goal of the corresponding lowest level need in the first order in the queue. The corresponding lowest level goal and the need become the active motive;

therefore, the agent starts focusing on it first.

Whenever an active motive is changed, it means that the agent's active goal has changed. In such situations, the motivation activator warns the attention mechanism to focus on the new active goal. Focusing on a new active goal means that the agent must delay executing the current activity and start executing the new plan related with the new active goal.

As an example it can be considered that an agent talking with its friend. When the agent feels hunger, it focuses on eating. While focusing on the need for food, it stops talking with its friends. In other words, the agent postpones talking for a later time. After satisfying its hunger, the agent can go back and continue talking.

In this context, the attention mechanism enables the agent to focus on meeting the active motive. While executing certain actions to satisfy a particular need, it keeps the agent focused on that activity. To do so, it directs controlling mechanisms. While performing the actions, if the agent's active goal changes, the attention mechanism changes the focus of the controllers. In other words, the agent stops executing the action by warning its actuator and perception controllers. These components start focusing on the new active goal. To do so, the actuator delays the current action to continue after finishing the new active goal.

Another important responsibility of the attention mechanism is to notify the motivation activator to change the order of the needs. As mentioned in the proposed framework, strong emotions can change the order of the needs. To realise this idea whenever a strong emotion is generated, the attention mechanism informs the motivation activator to move the corresponding need up or down in the hierarchy. If the strong emotion is positive, then the corresponding need goes one level down in the hierarchy. Likewise, if the strong emotion is negative, then the corresponding need goes one level up in the hierarchy. If the hierarchy of the needs changes, then the queue which holds the goals in the hierarchical order is also updated accordingly.

The last component of the reactive layer is the actuator controller. This component provides the means to perform actions in the environment. To perform actions, the ReCau agent requires external components like a body, arms, or just a message passing mechanism. These additional mechanisms may vary according to the design purposes of an agent. If the purpose is to develop a humanoid robot, then it is required to have an actuator such as a body.

The actuator controller guides the execution of the tasks which are dispatched by the deliberative layer. The dispatcher in the deliberative layer gives detailed actions related with the intention of the agent to the attention mechanism. Then the attention mechanism sends the relevant data to the actuator controller. The actuator controller provides the means to accomplish actions to achieve the intention. The intentions of the ReCau agent are also held in an ordered queue. The first intention in the queue is executed first. The other intentions in the queue wait until the corresponding goal is activated.

5.3 Deliberative Layer

The deliberative layer provides the means for learning, planning, conflict resolution with other agents, and dispatching the tasks to the components in the lower layer. The deliberative layer is the slowest layer of the ReCau. It employs the reinforcement learner, the discrete feasible planner with the plan library, the manipulator with the belief base and the dispatcher.

Whenever a pre-condition, a goal and a need (i.e. a pre-condition and a motive) are sent to the deliberative layer, they first reach the reinforcement learner. The learning types in ReCau are supervised and unsupervised learning. This means that the ReCau agent can learn in an unsupervised manner by just observing the other agents or it can learn under supervision by receiving a plan directly from another agent.

These learning types are realised by reinforcement and the reinforcement learner is responsible for learning. To explain the functions of the reinforcement learner, first the planning structure of ReCau should be explained. Therefore, learning types employed in ReCau is explained later in this section.

The second component in the deliberative layer is the planner. Like in the other planning approaches, the basic ingredients of the ReCau planning approach are states, conditions, actions, time, a criterion and a plan. A state space captures all possible situations that could arise. In ReCau, the state space is defined discretely in such a way that the state space is defined by state conditions. There are three different types of state conditions in the ReCau architecture: (1) initial-state conditions, (2) transition-state conditions, and (3) goal-state conditions.

The pre-conditions sent from the reactive layer are called the initial-state conditions. A state condition is denoted by c corresponding to a Boolean expression involving one or more state variables. The detailed description of the state conditions is not given here, but it suffices to say that it will include expressions composed of Boolean connectives and comparisons such as "$x = 1 \wedge y \leq 10$" where x and y are two of the state variables. Let C represent the set of all Boolean expressions defined over a given set of state variables $\vec{x}$.

The planning problem involves starting in an initial-state condition $c_0 \in C$ and trying to arrive at a goal-state condition $c_g \in C$. In the proposed approach, a ReCau agent at a given initial-state condition tries to reach the goal-state conditions which meet its goal to satisfy the corresponding need. The actions which manipulate the state conditions are selected in a way that tries to achieve a particular goal. In the ReCau architecture, each action taken results in achieving either a transition-state condition or a goal-state condition.

Time in ReCau is implicitly modelled by reflecting the fact that actions follow in succession. The particular notion of time is not important, but the proper sequence of actions is maintained. The desired outcome of a plan in terms of the state conditions and actions is called a criterion. The criterion in our approach is feasibility not optimality. Regardless of the efficiency, a ReCau agent tries to find plans that lead to arrival at a goal-state condition that is meeting the goal.

A plan imposes a specific strategy or behaviour on a ReCau agent. The plans simply specify a sequence of actions to be taken. With this respect, ReCau plans consist of five major parts: (1) an initial-state condition, (2) actions, (3) a sequence of actions (4) transition-state conditions, and (5) a goal-state condition. Each plan corresponds to a particular need and a goal-state condition with a mean value of a satisfaction degree (μ) and a variance value (σ^2). The possible plan trajectories for a particular scenario are illustrated in Figure 5.4.

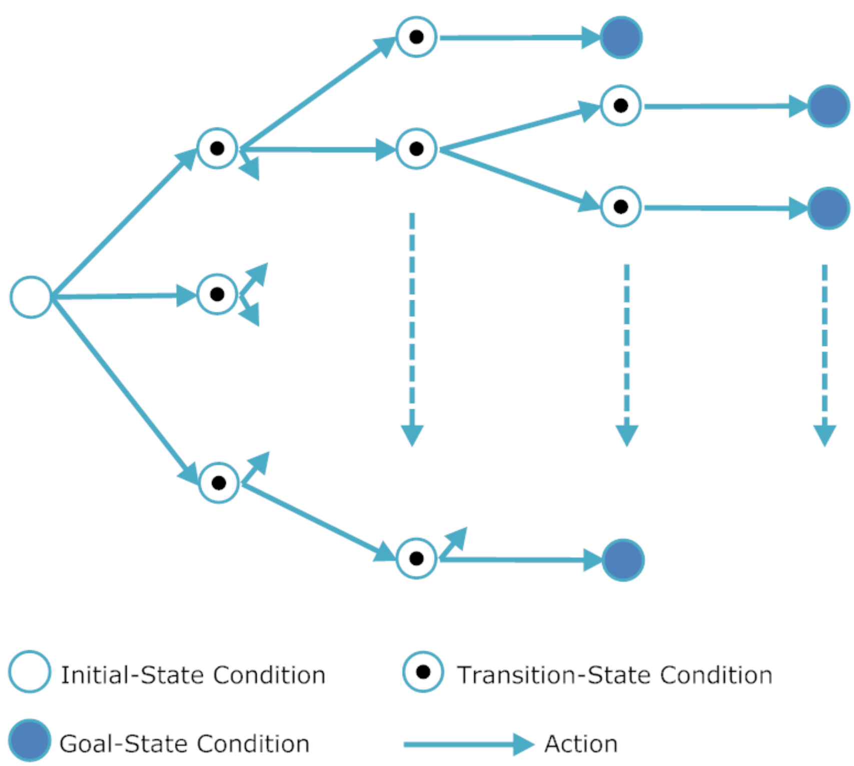

Figure 5.4: A Possible Plan Trajectory in ReCau

To illustrate the basic planning mechanism in ReCau, we adopt the discrete feasible planning model provided by LaValle [140].

Each distinct situation for the world is called a state. The set of all possible states is called a state space, S. As mentioned above, a state s is discretely defined through the different combinations of values for the state variables, $\vec{x}$, such that $s = \langle u_1, u_2, ..., u_n \rangle$ where $x_i = u_i$ for $1 \leq i \leq n$. The value u_i comes from the domain of the variable x_i (for instance, for a Boolean variable the corresponding domain is the set $\{true, false\}$).

By applications of the actions of the ReCau agents, the world (environment) is transformed. Each action, a, manipulates the values of (some of) the state variables, $x_i = u_i$, so that it yields new values for the state variables, $x_i = u'_i$. By this transformation, the current state, $s = \langle u_1, u_2, ..., u_n \rangle$, results in a new state, $s' = \langle u'_1, u'_2, ..., u'_n \rangle$. This transformation of state variables is specified by a transition function on state conditions, f, by specifying the pre- and post-conditions of the actions that can be taken. As far as actions are concerned, it is needed to know if an action can be applied at a given state and what effect it will have on that state.

Let $A(c)$ denote the action space for each condition c, which represents the set of all actions that could be taken when the condition c is satisfied. Different actions with the same pre-condition c may naturally have different post-conditions, possibly resulting in different states. For any given action $a \in A(c)$ where c is the pre-condition of the action a and c' the post-condition, the state condition transition equation can be shown as follows:

$$c' = f(c, a) \tag{5.1}$$

The set A of all possible actions (action repertoire) over all conditions can be defined as shown in Equation 5.2.

$$A = \bigcup_{c \in C} A(c) \tag{5.2}$$

It must be noted that all conditions are members of the condition space, $c \in C$. Not all the conditions will necessarily have associated actions.

A ReCau agent comes equipped with an initial action repertoire depending on its design purpose which can be expanded by reinforcement learning. Note that actions can be described in a suitable framework based situation calculus such as that of Reiter's [189] which is consistent with the adopted model.

The states satisfying a particular need are the ones meeting a particular goal-state condition, c_g, for example, the condition "`switch` $=$ *true*" is satisfied by any state where the light switch (represented by the variable `switch`) is turned on, regardless of the values of the remaining state variables. The set of goal states which satisfy a particular need can be defined as $S_g \subseteq S$.

In ReCau, the task of the planner is to find a finite sequence of actions to achieve a goal state $s \in S_g$ which meets the given goal-state condition c_g from the given initial-state condition c_0. Equipped with this formalism, the model of discrete feasible planning in ReCau adopted from Lavalle [140] is shown below:

1. A state space S, that is, a finite set of states defined over $\vec{x}$
2. An initial state condition c_0 and a goal state condition c_g
3. An initial state set $S_0 \subseteq S$ defined by $S_0 = \{s \in S \mid s \text{ satisfies } c_0\}$
4. A goal state set $S_g \subseteq S$ defined by $S_g = \{s \in S \mid s \text{ satisfies } c_g\}$
5. For each condition $c \in C$, a finite action space $A(c)$
6. A condition transition function f that produces a state condition $f(c, a) \in C$ for every $c \in C$ and $a \in A(c)$

The planner is responsible for developing plans based on the model of discrete feasible planning. Whenever an initial-state condition is received from the lower layer, the ReCau agent develops plans by the help of the plan library. The entity relationship diagram of the plan library is shown in Figure 5.5

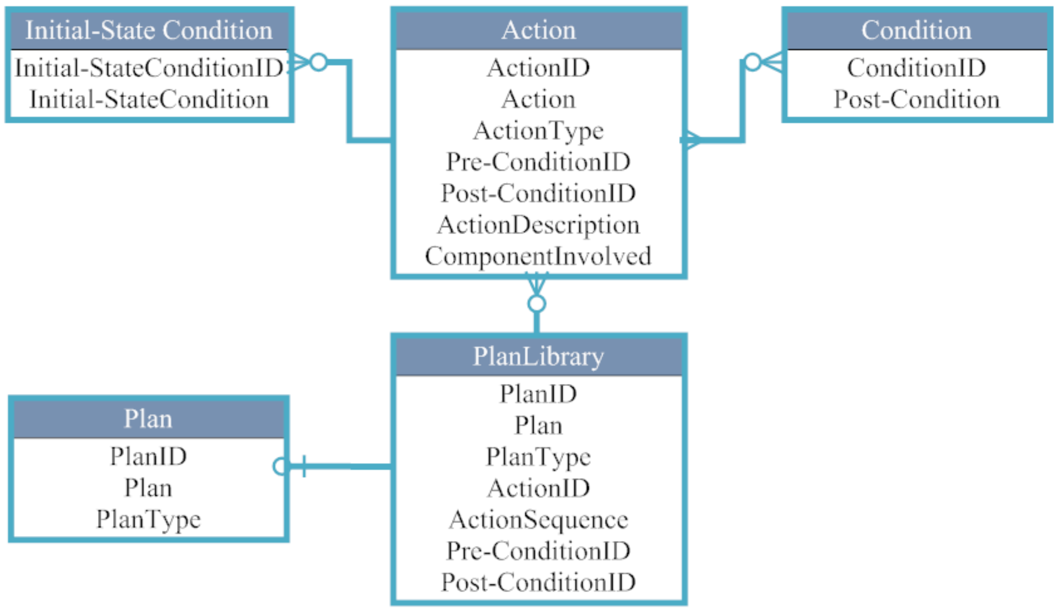

Figure 5.5: The Entity-Relationship Diagram of the Plan Library

The planner simply constructs plans by using a depth-first search planning algorithm. A general depth-first search planning algorithm is shown in Algorithm 5.3.

The algorithm starts off from the initial state condition c_0 (or equivalently from any state satisfying c_0), and by exploring the pre- and post-conditions of the actions that can be taken through the condition transition function f, searches a path using a stack data structure (represented by S) that would lead to the goal-state condition (or equivalently to any state satisfying c_g).

This algorithm is meant to illustrate the generation of only a single path (plan alternative) by depth-first search. In an implementation of the ReCau architecture,

Algorithm 5.1 Depth-First Search Planning Algorithm of ReCau

```
S.Push(c_0)
while S ≠ ∅ do
  c ← S.Pop()
  Mark c as visited
  if c is satisfied by any s ∈ S_g then
    return Success
  else {c is not satisfied by any s ∈ S_g}
    for all a ∈ A(c) do
      c′ ← f(c, a)
      if c′ not visited then
        S.Push(c′)
      else {c′ is visited}
        Resolve duplicate c′
      end if
    end for
  end if
end while
return Failure
```

this algorithm may be used to develop all plan alternatives based on a given initial-state condition. Other search algorithms can also be used if there are additional requirements and/or constraints.

In the frame of the given explanation on the planning approach adopted in the architecture, subsequently learning types in ReCau are explained.

Whenever a goal is received by the deliberative layer, the reinforcement learner first checks if the goal is to "slearn" or "ulearn". In the ReCau architecture, a goal can be "slearn" if the ReCau agent receives a plan from another ReCau agent. A goal can be "ulearn" if a ReCau agent sees that another ReCau agent is executing a plan. While the "ulearn" means the goal of the agent is to learn in an unsupervised manner, the "slearn" means the goal of the agent is to learn in a supervised manner.

If the goal is to "slearn", the ReCau agent starts supervised learning. While starting supervised learning, the ReCau agent first checks if the received plan is completely known. If the plan is known completely, then the ReCau agent stops learning. Otherwise, the ReCau agent attempts to find the unknown actions in the received plan and the known actions in the plan are filtered out.

Then the ReCau agent attempts to find the teacher agent's place in the social context. The social context of the ReCau agent is explicitly defined in the reinforcement learner. This means that each agent knows the family of itself, the agents who has influence on itself, the agents who have similarities to it and the other agents. Accordingly, the learner agent determines the teacher agent's place in the social context. It must be noted that the social context of each agent can be different from the other agents.

Afterwards, the ReCau agent writes the unknown actions with their pre- and post- conditions to the plan library. While writing these actions, the mean value of the satisfaction degree (μ) and the variance value (σ^2) corresponding to the teacher agent's place in the social context are assigned to the goal-state condition. For this purpose, the following social context and the corresponding values can be adopted in the implementation.

The Family of the Agent: $E(\mu) = 0.70$, $E(\sigma^2) = 0.05$

The Agents with Influence: $E(\mu) = 0.60$, $E(\sigma^2) = 0.05$

The Agents with Similarities: $E(\mu) = 0.50$, $E(\sigma^2) = 0.10$

The Other Agents: $E(\mu) = 0.40$, $E(\sigma^2) = 0.10$

Within this frame, the flowchart of the supervised learning process is shown in Figure 5.6.

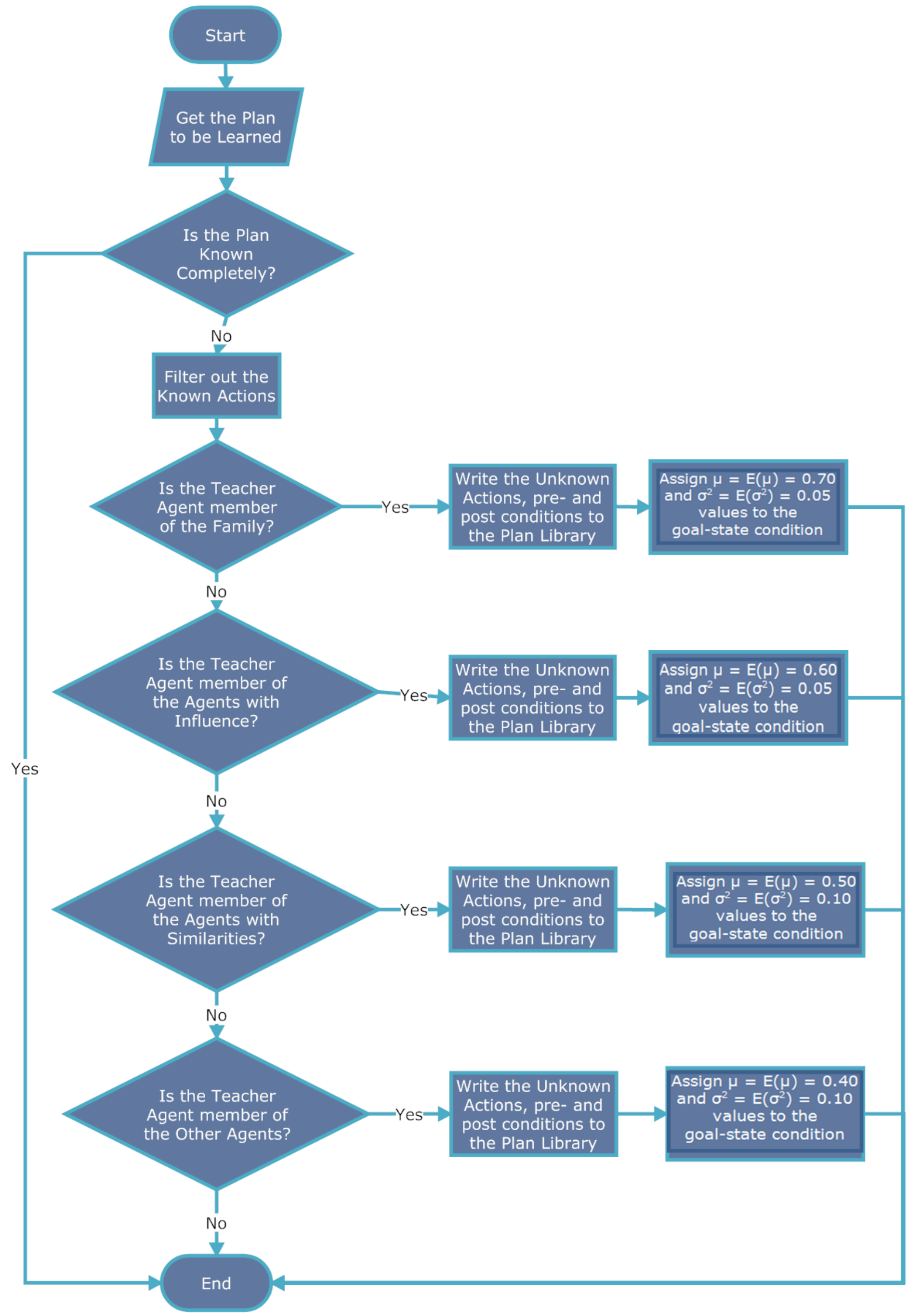

Figure 5.6: The Flowchart of the Supervised Learning Process

If the goal is to "ulearn", the ReCau agent starts unsupervised learning. When starting unsupervised learning, to provide reinforcement in this approach, the ReCau first assigns an expected mean value of the satisfaction degree and an expected variance value to the plan that is going to be learned. These values signify the expectations of the ReCau agent. To do so the ReCau agent considers the teacher agent in the social context given previously.

As seen in this context, each level corresponds to a mean value of the satisfaction degree $E(\mu)$ and a variance value ($E(\sigma^2)$). In unsupervised learning these values are considered as the expected mean value of satisfaction degree and the expected variance value to the plan that is going to be learned.

After defining the expected values, the ReCau agent learns plans in accordance with the conditions and the actions observed. The policy maps the state conditions of the world to the actions that the agent ought to take in those states. To illustrate reinforcement learning, here the policy is considered as a plan.

To realise this idea, "reward" (r) and "punish" (p) values are introduced. The reward is $r = 0.20$ and punish is $p = -0.20$ by default. By using these values, after receiving each action the learner agent recalculates the expected mean value of the satisfaction. If the action is not known, it is rewarded by multiplying $1 + r$ and $E(\mu)$. If the action is known, it is punished by multiplying $1 + p$ and $E(\mu)$. The outcome of the multiplication becomes the new expected mean value of the satisfaction degree.

This process continues for each action observed until either the agent reaches to a goal-state condition or the expected mean value of the satisfaction degree becomes less than 0.40 which is the learning satisfaction limit (ς_l^2). If the agent reaches to the goal-state conditions the last calculated expected mean value of the satisfaction is written as the mean value of the satisfaction degree of the goal-state condition. If the mean value of the satisfaction becomes less than the learning satisfaction limit, the ReCau agent stops the learning process.

Within this framework, the flowchart of the supervised learning process can be shown in Figure 5.7.

The actions learned by using these types of learning can be put in practice by the agent later on; since, the reinforcement learner learns plans and deploys them in the plan library.

In summary, if the goal sent from the reactive layer is related to learning, then the agent starts learning. Otherwise, it develops plan alternatives which meet the goal received by the planner. Then the pre-condition, the fully developed plan alternatives with the mean values of the satisfaction degrees and variance values and the corresponding need and goal are sent to the manipulator.

The responsibilities of the manipulator are to update conditional pro-attitudes, judge alternatives and resolve conflicts between other agents by using the belief base. The conditional pro-attitudes that may include pro-attitudes like obligations, likes, dislikes are explained in terms of beliefs; therefore, they are stored in the belief base.

When the manipulator receives a pre-condition, if it is required, the conditional pro-attitudes are changed in the belief base. Some pre-conditions require taking some actions but at the same time they may result in changing the pro-attitudes of the agent due to changing conditions. Therefore, the pro-attitudes are changed by the manipulator whenever required.

After making changes in the pro-attitudes, the manipulator simply checks the post-conditions of each plan alternative to see if there is an associated conditional pro-attitude. Whenever the manipulator finds a conditional belief (i.e., a conditional pro-attitude) associated with a post-condition, it analyses the impact of the conditional belief on that plan alternative.

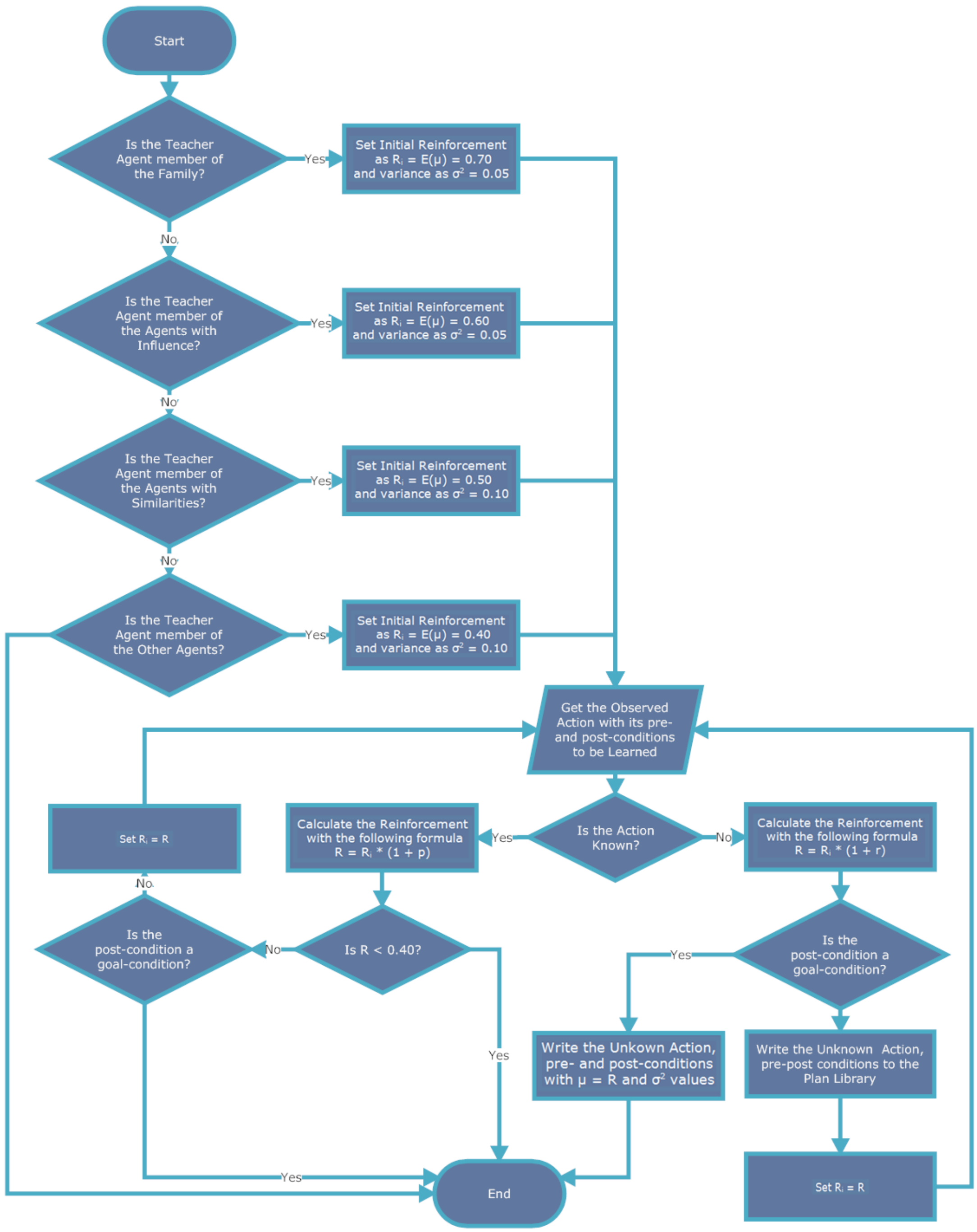

Figure 5.7: The Flowchart of the UnSupervised Learning Process

To realise this impact, each conditional pro-attitude has a certain impact factor (ψ). These impact factors can be positive or negative values. While a positive impact factor increases the mean values of satisfaction degrees, a negative impact factor decreases the mean values of satisfaction degrees. In the ReCau architecture, the impact factors are real numbers between -1 and 1. The factor value of 0 signifies no influence, while the value 1 and -1 signifies the strongest influence.

By using these impact factors, the manipulator simply recalculates the corresponding mean values of the satisfaction degrees of the plan alternatives. The recalculated mean values are called the ameliorated mean values of the satisfaction degrees ($\tilde{\mu}$). They are calculated as shown in Equation 5.3.

$$\tilde{\mu} = \begin{cases} 1 & \texttt{if } \mu + (\mu \times \psi) \geq 1 \\ \mu + (\mu \times \psi) & \texttt{if } 0 \leq \mu + (\mu \times \psi) < 1 \end{cases} \tag{5.3}$$

After finding the ameliorated mean values of the satisfaction degrees, these values are assigned to the corresponding plan alternatives as their new mean values of the satisfaction degrees ($\mu = \tilde{\mu}$).

Afterwards, all of the plan alternatives with their corresponding mean values and variance values and the corresponding need are sent to the causal layer.

The last responsibility of the manipulator is to enable an agent to collaborate on pursuing common needs by resolving conflicts with other agents. When several ReCau agents have common needs, they will adjust unless they have contradicting pro-attitudes like beliefs, likes, dislikes and obligations. A ReCau agent is harmonious with other ReCau agents while trying to satisfy their common, non-conflicting needs.

The negotiation process is managed by the manipulator to collaborate on common needs. When an agent suggests a plan to a ReCau agent to collaborate on pursuing a common need, the ReCau agent tends to accept the plan alternative if it does not have contradicting pro-attitudes. If the ReCau agent has contradicting pro-attitudes, then it offers its plan alternative. If the other agent insists on its own plan alternative by providing additional conditional pro-attitudes, the ReCau agent considers this additional conditional pro-attitude. In other words, the ReCau agent first checks if the new conditional pro-attitude has a stronger influence and then adjusts itself to collaborate.

The decision-making process is managed by the causal layer. Whenever the deliberative layer generates plan alternatives, it sends the plan alternatives to the causal layer. In particular, the agent sends plan alternatives with the mean values of the satisfaction degrees and the variance value and the corresponding need to the causal layer. The decision is made in the causal layer and the intention is determined by selecting one plan among the received plan alternatives. In this layer, an emotion is also determined.

After determining both the emotion and the intention, they are sent back to the deliberative layer specifically to the dispatcher. The function of the dispatcher is to assign tasks to the components in the reactive layer. To assign those tasks, according to the intention (i.e., the selected plan) and the emotion, the dispatcher obtains the details of the plans from the plan library. In the plan library, each action is described explicitly in such a way that each action corresponds to certain components.

Moreover, in ReCau, the emotions are also kinds of plans; therefore, they are held in the plan library. To realise the affect display, the plan library also contains fully developed action sequences related with regular and strong emotions. In accordance with the emotion and the intention of the agent, tasks are formed by the dispatcher. Finally, these tasks are sent to the attention mechanism to direct the corresponding components for execution.

5.4 Causal Layer

The decision-making process and emotion generation process are managed by the causal layer. Whenever the deliberative layer generates plan alternatives, it sends them to the causal layer. In particular, a ReCau agent sends the plan alternatives with the mean values of the satisfaction degrees and the variance values and the corresponding need to the causal layer. This data first reaches the decision-making mechanism.

The decision-making mechanism uses these values to determine the satisfaction degrees for each plan alternative. Here the aim is to obtain satisfaction degrees (ς). These satisfaction degrees are normally distributed random values that are generated by using the corresponding mean values of the satisfaction degrees and the variance values of the plan alternatives. In this manner, it is aimed to provide action flexibility (or a semi-autonomous action capability) to a ReCau agent.

To obtain satisfaction degrees, first the polar technique is applied. The polar technique is a modified form of the Box-Muller method [32]. The polar technique is shown in Algorithm 5.4.

Algorithm 5.2 Polar Technique Algorithm

repeat
  $U_1 \Leftarrow \texttt{iid}\ U \sim [0,1]$
  $U_2 \Leftarrow \texttt{iid}\ U \sim [0,1]$
  $V_1 \Leftarrow 2 \times U_1 - 1$
  $V_2 \Leftarrow 2 \times U_2 - 1$
  $W \Leftarrow V_1^2 + V_2^2$
until $W < 1$
$Y \Leftarrow \sqrt{-2 \times \ln(W) \,/\, W}$
$\rho_1 \Leftarrow Y \times V_1$
$\rho_2 \Leftarrow Y \times V_2$

By using this algorithm, initially two independent and identically distributed (iid) uniform random variables (U_1 and U_2) are generated. Then these uniform random variables are transformed into two normally distributed random numbers (ρ_1 and ρ_2). These are independent and identically distributed (iid) and normally distributed with the mean value of 0 and standard deviation of 1 (ρ_1 and ρ_2 are iid $N \sim (0, 1)$).

After obtaining two normally distributed random numbers, we use one of them to calculate a satisfaction degree (ς). The formula for calculating satisfaction degrees is shown in Equation 5.4.

$$\varsigma = \begin{cases} 1 & \texttt{if } \rho_1 \times \sigma + \mu \geq 1 \\ \rho_1 \times \sigma + \mu & \texttt{if } 0 < \rho_1 \times \sigma + \mu < 1 \\ 0 & \texttt{if } 0 \geq \rho_1 \times \sigma + \mu \end{cases} \tag{5.4}$$

In this formula, μ is the mean value of the satisfaction degree and σ is the standard deviation. The standard deviation is found by the formula given in Equation 5.5.

$$\sigma = \sqrt{\varsigma} \tag{5.5}$$

As stated previously, each plan alternative corresponds to a different mean value of the satisfaction degrees and different variance values. Therefore, for each plan alternative a different satisfaction degree is to be calculated.

By applying the polar method and calculating the satisfaction degrees for each plan alternative, the ReCau agent selects the most satisfactory alternative. The most satisfactory plan alternative is the one with the highest satisfaction degree. By selecting one plan among all the received plan alternatives, the agent determines its intention. In other words, the agent makes decision by selecting one of the plan alternatives. The selected plan alternative becomes the intention of the agent.

After determining the intention, the next task is to generate an emotion. Each action taken does not guarantee emotion generation. If the satisfaction degree is above or below a certain limit, the emotion generation mechanism generates an emotion in

Table 5.1: Default Emotion Limit Values

Strong Positive Emotion Limit	0.90
Positive Emotion Limit	0.80
Negative Emotion Limit	0.20
Strong Negative Emotion Limit	0.10

accordance with the received need. In the ReCau architecture, if the satisfaction degree is above the positive emotion limit, it generates a regular positive emotion. If the satisfaction degree is above the strong positive emotion limit, it generates a strong positive emotion. Likewise, there are the negative emotion limit and the strong negative emotion limit. If the satisfaction degree is lower than these limits, the negative emotions are generated.

The default emotion limits in ReCau is given in Table 5.1.

To realise emotion generation in ReCau, in the emotion generation mechanism each need is associated with a positive emotion, a strong positive emotion, a negative emotion and a strong negative emotion. The entity-relationship diagram of the emotions is shown in Figure 5.8.

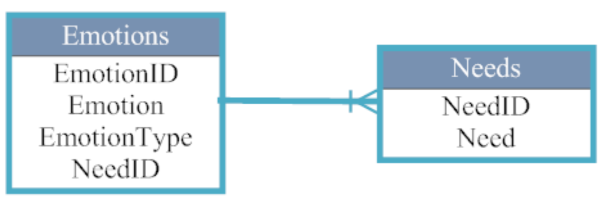

Figure 5.8: The Entity-Relationship Diagram of the Emotions

Whenever a need is sufficiently satisfied or inadequately satisfied, the corresponding emotion is generated. In the implementation of the ReCau, the needs and the emotions provided in the previous section can be adopted. It must be noted that, more needs and emotions can be introduced in the implementation.

After determining both the emotion and the intention, they are sent back to the deliberative layer, specifically, to the dispatcher.

5.5 Comparison with Existing Architectures

In this section of the study, the proposed architecture ReCau is compared with several existing architectures. The most distinctive attributes of ReCau are employing motives and displaying affect. Therefore, ReCau is only compared with existing architectures which employ motives and/or display affect.

The first architecture that is compared with ReCau is Tok [23]. This architecture is developed to provide affect display and social behaviour capabilities to an agent. The Tok architecture provides the means to show a goal directed behaviour. However, it does not employ any motives to generate goals. The emotions that are available in the Tok architecture are mostly limited with basic emotions like fear and anger. However, ReCau employs motives and more complex emotions. In addition, Tok does not adopt any learning approaches.

The other architecture that is compared with ReCau is MPA [24]. MPA is an instance of PRS with enhanced features. PRS is the most commonly adopted hybrid architecture employing the belief, desire and intention approach like ReCau. PRS and ReCau employ a plan library and include explicit representations of beliefs, desires, and intentions. The plan library in both architectures contains a set of plans. In PRS, the set of plans can be activated in a goal- or data- driven fashion. In ReCau, the set of plans can be activated in a motivation driven fashion which means that while trying to satisfy a certain need, the plans are activated to make decision. With this respect, a ReCau agent is driven by its needs. This provides a higher level of autonomy. One another important difference between these architectures is that ReCau enables an agent to simulate emotional states.

The first enhanced feature of MPA is the busyness filter. This filter checks the load on the planner and filters out goals if the planner is busy. ReCau also has a filtering mechanism which enables the planner to focus on only one issue at a time and reduces the load of the planner. The filtering mechanism in ReCau also allows the system to work in accordance with resource boundedness by filtering out data irrelevant to the needs of an agent.

MPA has a management process to provide meta-management controls. This process decides actions that are going to be performed next in good computational time. For this purpose, the management process controls every lower level action. ReCau has no such layer. But the motivation activator provides the same efficiency without introducing an additional layer. Finally, like PRS, MPA does not support the generation of emotions.

Another architecture that has similarities with ReCau is Motivated Agency [172]. In MA, motives are used to generate motivated goals. According to this architecture, firstly the agents pursue goals with a higher motivation level. The motivation generation structure of Motivated Agency architecture is an instance of ideas of Walter B. Cannon [45]. However, ReCau adopts the theories of needs. The motives in Motivated Agency are not explicitly defined. However, ReCau agents try to satisfy explicitly defined lower level needs first. Moreover, being driven by needs is realised by only employing a motivation activator. Unlike ReCau, Motivated Agency does not simulate affect display.

CMattie is another architecture that enables an agent to display emotional states [158]. CMattie learns more complex emotions from its experiences. It pursues goals reinforced by the emotional valence. The approach adopted in CMattie is different form classic reinforcement in which a positive or negative valence is provided through feedback. CMattie architecture enables an agent to pursue the greatest pleasure and avoid displeasure. ReCau also has a similar mechanism to pursue favourable situations while refraining from unsatisfactory situations. The most distinguishing difference between these architectures is that ReCau employs motives to generate goals.

Another architecture that adopts emotional reasoning is Èmile. Èmile adopts social learning theory like ReCau. However, it does not employ motives. Èmile agents are capable of recognising the plans of other Èmile agents. Like ReCau, the planning approach adopted in Èmile is not effective.

Another architecture that is compared with ReCau is H-Cogaff [211, 210]. Both architectures employ reactive and deliberative layers. But the upper most layers are significantly different. The highest layer of H-Cogaff is a meta-management layer which controls and monitors the reactive and deliberative layers. In ReCau, there is no such controlling layer. The upper most layer of ReCau is the causal layer which supports decision-making and emotion generation.

In H-Cogaff some motives are introduced to provide a filter to focus attention. In this architecture, the attention mechanism selects motivators to attend to. Similarly, a ReCau agent filters observed data according to its needs. Moreover, these needs also provide the means to focus attention. While performing similar activities, the underlying mechanisms are totally different in these architectures. Moreover, in ReCau motives are explicitly defined as needs.

The major difference between H-Cogaff and ReCau is related with emotion generation. H-Cogaff distinguishes emotions in three categories: primary, secondary and tertiary emotions. Besides, each emotion type is related with one layer of the architecture. However, emotions in ReCau are related to the satisfaction/dissatisfaction of needs. By following ERG, ReCau employs three levels of needs. Accordingly, the existence needs correspond to basic emotions, while relatedness and growth needs correspond to non-basic emotions. In addition to this, the relatedness needs correspond to social emotions, while growth needs correspond to non-social emotions.

EMAI is another architecture that is capable of affective decision-making [18]. It makes decisions through emotional appraisal. Based on their emotianl states, EMAI agents can change their behaviour. It also includes a motivational drive generation component to motivate an agent emotionally. This architecture does not support the development of social agents and does not employ motives.

Even though there are significant differences, CLARION has many similarities with ReCau. Both architectures adopt reinforcement learning. However, a CLARION agent learns from its experiences, while a ReCau agent learns from other agents by adopting social learning theory.

These two architectures enable agents to be driven by their needs. CLARION offers two different types of drives: primary drives and secondary drives. When explaining primary drives, CLARION employs the hierarchy of needs. Moreover, CLARION employs derived drives called secondary drives. These drives that can change over time are acquired in the process of satisfying primary drives [216]. Instead of this approach, ReCau adopts the ERG model which solves the overlapping problem in the hierarchy of needs. Another difference is that ReCau explains emotions as the result of satisfaction or dissatisfaction of needs. However, CLARION does not address this aspect of intelligence.

Another architecture that enables agents to show emotionally influenced behaviour is COGNITIVA. Similar to ReCau, COGNITIVA has reactive and deliberative layers. Besides, COGNITIVA includes a social layer to show social behaviour [117]. Social capabilities of ReCau are introduced without an additional layer. Instead of social layer, ReCau contains a causal layer. Even though, COGNITIVA does not employ any learning approaches and motives, it supports the development of believable agents.

When compared to LMRB, ReCau covers nearly all of the cognitive processes listed by Wang et. al. [233]. However, ReCau has only three layers. ReCau does not support a few higher cognitive processes like creation. However, ReCau can still generate human-like intelligent behaviour even if it is not inspired from the structure of the human brain.

The model of perceptual processes proposed by Wang [232] are similar to the processes adopted in ReCau. ReCau also categorises emotions as positive and negative. It hierarchically lists emotions as regular and strong emotions. However, in ReCau basic emotions are not associated with more complicated emotions. In ReCau emotions are associated with the hierarchy of needs. According to Wang's model motivation is triggered by an emotion or external stimulus. In ReCau, motivation is triggered by an external or internal stimulus not by an emotion. ReCau is motivation/need driven and only strong emotions have effect on the order of needs in the hierarchy.

When compared to AAS [231], ReCau agents are motivation driven not goal driven. Driven by the needs, external and internal stimulus triggers a ReCau agent to generate goals which are associated with its needs.

Last but not the least important issue is related with the flexibility attribute. All of the architectures discussed here are capable of performing certain actions. In their current form, these architectures consider an agent more like an individual system. Most of these architectures do not focus on cooperation among agents. However, ReCau attempts to enable cooperation with other ReCau agents to pursue common needs. For this purpose, ReCau agents are capable of resolving conflicts between them.

Chapter 6

Simulation Studies

In this chapter, two simulation studies are presented to illustrate the proposed architecture. The reactive and deliberative layers of the ReCau are in their classical form. The most distinctive layer of the architecture is the causal layer. Therefore, the simulations are performed to demonstrate the features of the causal layer. The first simulation is performed to illustrate the action flexibility provided by ReCau. The second simulation called the radar task is performed to illustrate the decision-making mechanism of the architecture in more detail.

6.1 Simulation of Action Flexibility

To illustrate action flexibility in ReCau, three simple experiments are performed. In these experiments, an agent named Jack is considered. Jack is supposed to go to work every day. Assume that to go to the work the agent has two plan alternatives: (1) Take a bus to work or (2) Drive to work. Further assume that the satisfaction of driving his car to the work is higher than that of taking a bus. To illustrate these plan alternatives in ReCau, it is needed to define two plan alternatives:

- Plan 1: Take a bus to work

- Plan 2: Drive the car to work

It must be noted that in ReCau, the plans consist of a sequence of actions. Therefore, whenever flexibility in selecting a plan alternative is provided, it means that action flexibility is also provided to an agent.

While defining plan alternatives in ReCau, the mean values of satisfaction degrees and the variance values to each plan alternative are assigned separately. Since the agent's satisfaction of driving the car is higher, the mean value of the satisfaction degree for Plan 1 is defined to be higher than that of Plan 2. Assume that these values are defined as follows:

- Plan 1: ($\mu_1 = 0.80$) and ($\sigma_1^2 = 0.05$)
- Plan 2: ($\mu_2 = 0.60$) and ($\sigma_1^2 = 0.05$)

Further assume that there are no conditional pro-attitudes associated with these plan alternatives; therefore, the ameliorated mean values are equal to the mean values (i.e. $\mu_1 = \tilde{\mu}_1 = 0.80$ and $\mu_1 = \tilde{\mu}_2 = 0.60$).

By applying Algorithm 5.4 and Equation 5.4 by using the mean values of the satisfaction degrees and the variance values above, $1,000$ satisfaction degrees (ς) are generated for the plan alternatives separately. In other words, it is attempted to understand how the agent Jack behaves when he goes to work. To understand it, the state in which the agent needs to go to work $1,000$ times is generated. Probability density functions (pdf) of these generated satisfaction degrees of the plan alternatives are shown in Figure 6.1.

As it can be seen in Figure 6.1, if the agent comes across with the state in which he needs to go to work, he tends to drive to work; since, the probability density functions intersect around $\varsigma = 0.70$. If the satisfaction obtained from the first plan alternative becomes lower than 0.70, the agent might unpredictably select the second plan alternative. In such cases, the agent takes the bus to work.

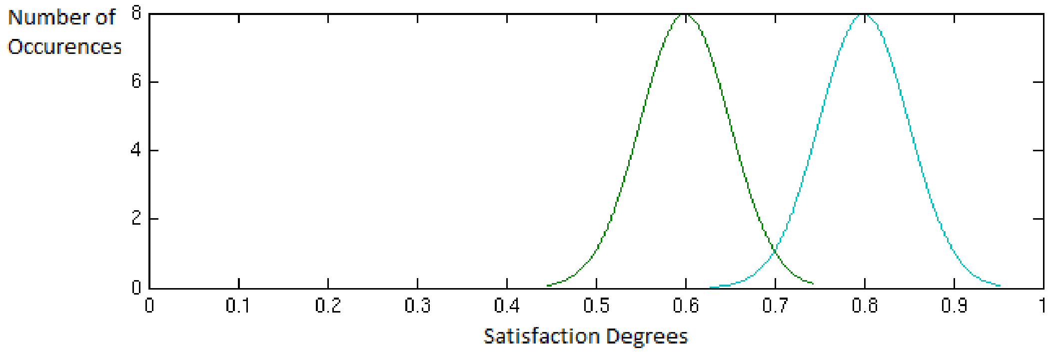

Figure 6.1: The Plot of the Probability Density Functions of Jack's Satisfaction Degrees in Experiment 1

As stated by Agre, even if agents have several different plan alternatives, most of the everyday activities are routine [5]. Therefore, in most cases they do not show unpredictable behaviours. In ReCau, variance values assigned to the plan alternatives provide means to support the ideas of Agre. The routine activities are activities such that they show too little variations. Therefore, for the given example if the variance value of the first plan alternative is reduced to $\sigma_1^2 = 0.005$ while the other values remains the same (i.e. $\mu_1 = \tilde{\mu}_1 = 0.80$, $\mu_2 = \tilde{\mu}_2 = 0.60$ and $\sigma_2^2 = 0.05$), then the first plan alternative would become a routine activity. In Figure 6.2, the results of the second experiment are shown. In the second experiment only variance value of the first plan is reduced while the other values remained the same.

As it can be seen in the figure, two pdfs do not intersect with each other and the satisfaction obtained from driving to work is always higher than taking the bus. Therefore, the agent will always drive to work routinely. In other words, Plan 1 has became a routine activity.

Although everyday activities are mostly routine, in some cases the agents can choose not to do their routine activities and to do alternative activities due to changing conditions. For instance, assume that the agent Jack goes to work with his car

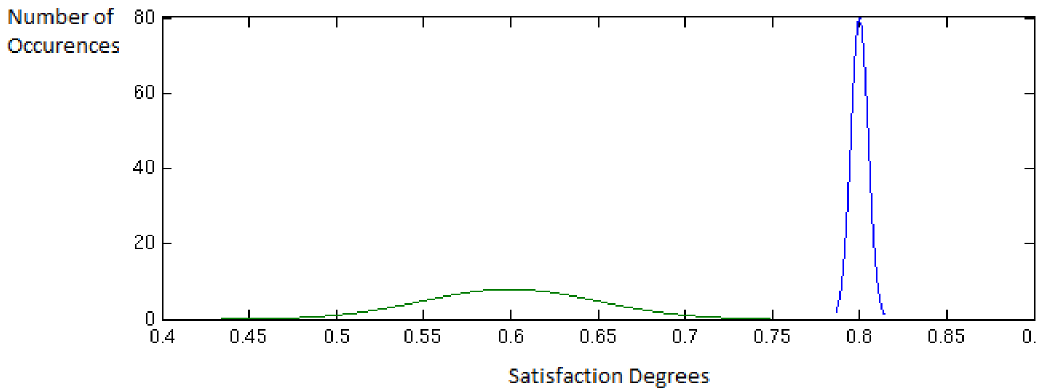

Figure 6.2: The Plot of the Probability Density Functions of Jack's Satisfaction Degrees in Experiment 2

routinely. But one day, he may choose to go to work by bus due to heavy weather conditions. If the agent believes that there is a heavy snow, then his satisfaction obtained from driving the car would reduce. Therefore, the agent would choose to go to work by bus.

To illustrate such situations in the proposed approach, conditional pro-attitudes are employed. For this case, assume that the pro-attitude of Jack is "There is a heavy snow." In other words, the agent believes that there is a heavy snow. As stated before, in the proposed approach such pro-attitudes have an impact over associated plan alternatives. For this instance, this pro-attitude has an impact over the first plan alternative.

To illustrate this idea, the impact factors (ψ) are assigned to the conditional pro-attitudes. For the given example, assume that $\psi = -0.80$ is assigned to the conditional pro-attitude "There is a heavy snow.". Since this pro-attitude is associated with the first plan alternative, it is needed to calculate the ameliorated mean value of the satisfaction degree of plan alternative 1 by applying Equation 5.3. Then the ameliorated mean value is obtained as $\tilde{\mu}_1 = 0.16$. The ameliorated mean value for plan alternative 2 remains the same; since, there are no other associated pro-attitudes.

Then the ameliorated mean value of the satisfaction degree is assigned as the new mean value of the satisfaction degree of the plan 1 $\mu_1 = \tilde{\mu}_1 = 0.16$. Once again while the other values remain the same ($\mu_2 = 0.60$, $\sigma_2^2 = 0.005$ and $\sigma_2^2 = 0.05$), Algorithm 5.4 and Equation 5.4 can be applied to generate the satisfaction degrees for $1,000$ times. The probability density functions of the third experiment is shown in Figure 6.3.

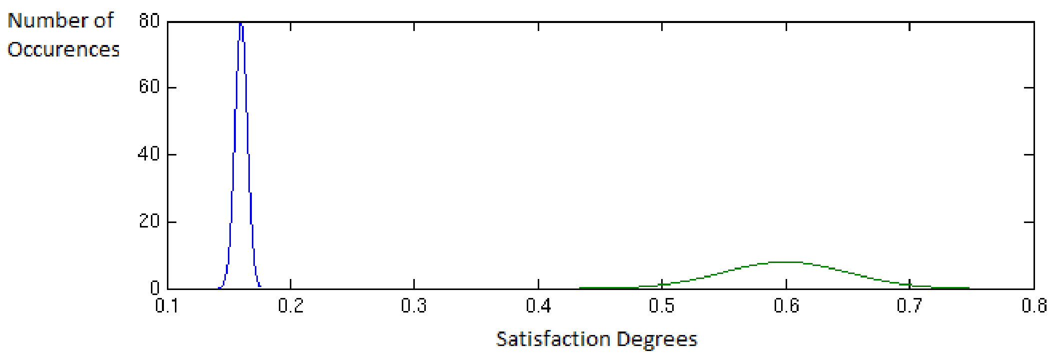

Figure 6.3: The Plot of the Probability Density Functions of Jack's Satisfaction Degrees in Experiment 3

As it can be seen in the figure, the routine activity became less satisfactory. Therefore, as long as the pro-attitude "There is a heavy snow" remains true, the agent takes the bus to work. Whenever the snow stops, the pro-attitude becomes false; therefore, the mean value of the satisfaction degree for the first plan alternative goes back to $\mu_1 = 0.80$. In other words, the agent would continue his routine activity by driving his car to work.

6.2 Radar Task Simulation

Radar task simulation is based on an organisational decision-making scenario. This simulation is undertaken to illustrate the decision-making mechanism of the Reactive-Causal Architecture. This simulation is undertaken by many other existing architectures; therefore, it provides the means to compare ReCau with some of those existing architectures. In this section, the radar task is described first. Then the former radar task simulation studies are reviewed. Then, the results of the radar task simulation obtained by employing ReCau agents are given. Finally, the results of the radar task simulation are evaluated.

6.2.1 The Description of the Radar Task

In the field of organisational research many researchers focused on determinants of organisational performance. For this purpose, organisational theorists attempted to develop various formalisms to predict behaviour. Several formal models are developed by using mathematics, simulation, expert systems and formal logic. Those models help researchers to provide information on organisational behaviour, determine errors and gaps in verbal theories, and determine if theoretical propositions are consistent [48].

Carley et. al. [48] explain organisational performance as a function of the task performed. A typical task is a classification choice task in which decision makers gather information, classify it and make a decision based on the classified information. In the field of organisational design, many researchers adopted the radar task in order to determine the impact of cognition and design on organisational performance. In this task organisational performance is characterised as accuracy.

In the radar task, the agents try to determine whether a blip on a radar screen is a hostile plane, a civilian plane, or a flock of geese. Originally, there are two types of radar tasks. The first one is static and the second one is dynamic radar task. In the static version of the task, the aircrafts do not move on the radar screen. In the dynamic radar task, the aircrafts move and the analysts may examine the aircraft for several times [147]. In the present study, the static version of the radar task is adopted. Therefore, the static radar task is explained below.

In this task, there is a single aircraft in the airspace at a given time. The aircrafts are uniquely characterized by nine different characteristics (features). The list of these features are shown in Table 6.1.

In the radar task simulations each of the above characteristics can take on one of three values (low = 1, medium = 2, or high = 3). A number of agents must determine whether an aircraft observed is friendly (1), neutral (2), or hostile (3). The number of possible aircrafts is 19,683 which is the number of different unique combinations of the features (3^9).

A task environment can either be biased or unbiased. If the possible outcomes of the task are not equally likely, it is said that the task environment is biased. If approximately one third of the aircrafts are hostile and one third of the aircrafts are friendly, then the environment is said to be unbiased. Lin and Carley [146] state that biased tasks are less complex; since, a particular solution is preponderate.

The true state of an aircraft is determined by adding the values of the above 9 features. In an unbiased environment, if the sum is less than 17, then the true state of the aircraft is friendly. If the sum is greater than 19, then the true state of the aircraft is hostile. Otherwise, the aircraft's true state is neutral. The true state of the aircraft is not known before making the decision [146].

Table 6.1: The Features of an Aircraft [Source: Lin and Carley [147]]

Name	Range	Categorisation of Criticality		
		Low	Medium	High
Speed	200-800 miles/hour	200-400	401-600	601-800
Direction	0-30 degrees	21-30	11-20	0-10
Range	1-60 miles	41-60	21-40	1-20
Altitude	5,000-50,000 feet	35k-50k	20k-35k	5k-20k
Angle	(-10)-(10) degrees	(4)-(10)	(-3)-(3)	(-10)-(-4)
Corridor Status	0(in), 1(edge), 2(out)	0	1	2
Identification	0(Friendly Military), 1(Civilian), 2(Unknown Military)	0	1	2
Size	0-150 feet	100-150	50-100	0-50
Radar Emition Type	0(Weather), 1(None), 2(Weapon)	0	1	2

The responsibility of the organisation is to scan the air space and make a decision as to the nature of the aircraft. Some of the agents (the analysts of the organisation) have access to information on the aircraft related to its features. Based on this information, the agents make decision and develop a recommendation whether they think the aircraft is friendly, neutral, or hostile. The recommendations are processed or combined in accordance with the organisational structure. The types of organisational structures are as follows [147]:

- Team with Voting: In this type of an organisational structure, each analyst has an equal vote. Each analyst examines information and makes a decision. This decision is considered as the vote of the analyst. The organisational decision is made by the majority vote. This structure is illustrated in Figure 6.4.

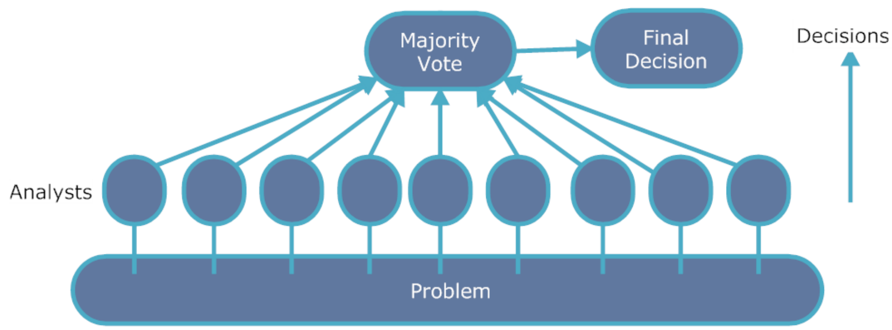

Figure 6.4: Organisational Structure of Team with Voting [Source: Lin and Carley [146]]

- Team with Manager: In this structure, each analyst reports its decision to a single manager. Like the team with voting, analysts examine information and recommend a solution. Based on these recommendations, the manager makes an organisational decision. The Team with Manager structure is shown in Figure 6.5.

- Hierarchy: In the hierarchical structure, each analyst reports to its middle-level manager and the middle-level managers report to the top-level manager. The analysts examine information and make recommendations. Then the middle-level managers analyse the recommendations from their subordinates and make a recommendation to the top-level manager. Based on the middle-level managers' recommendations, the top-level manager makes organisational decision. This structure is illustrated in Figure 6.6.

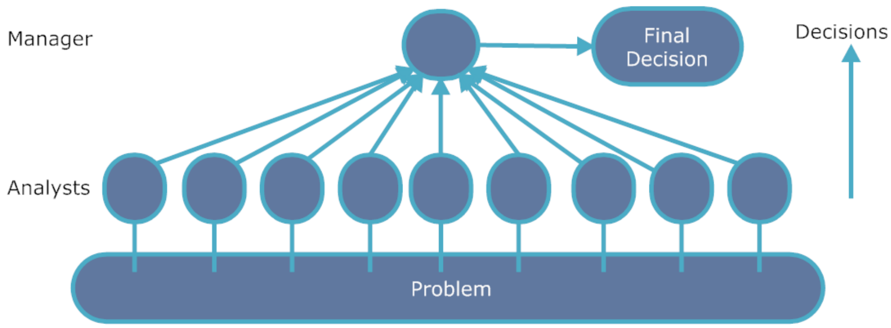

Figure 6.5: Organisational Structure of Team with Manager [Source: Lin and Carley [146]]

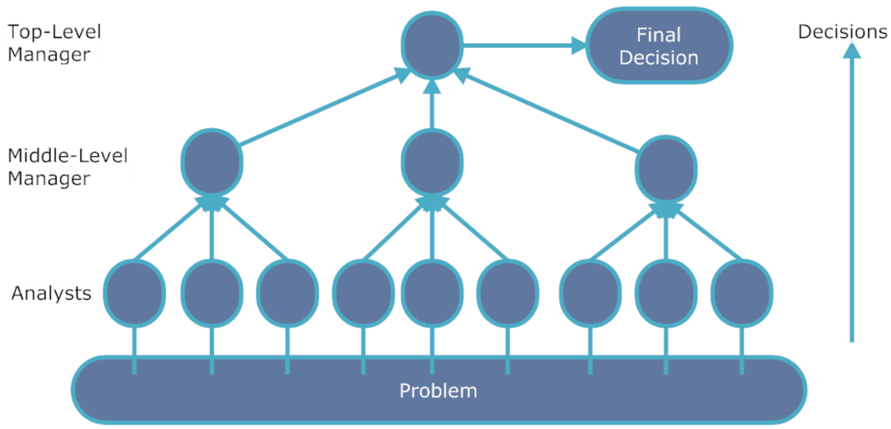

Figure 6.6: Organisational Structure of Hierarchy [Source: Lin and Carley [146]]

- Matrix: This structure is also a hierarchical structure. However, in this structure, each analyst reports to two middle level managers. Each analyst examines information and makes a recommendation. By examining the recommendations of their subordinates, the middle-level managers make a decision and report to the top-level manager. Top-level manager makes an organisational decision based on the recommendations of the middle-level managers. The matrix structure is shown in Figure 6.7.

As illustrated in the above figures, in the radar task simulations, each structure

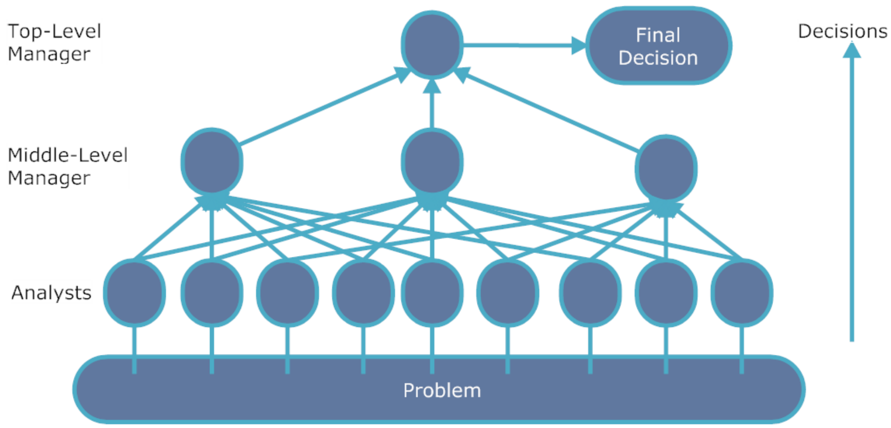

Figure 6.7: Organisational Structure of Matrix [Source: Lin and Carley [146]]

consists of nine analysts. As it can be seen, some structures also include middle and/or top-level managers.

Within an organisation, there are also resource access structures. These structures determine the distribution of information to the analysts. Each analyst may have access to particular characteristics. There are four different types of resource access structures:

- Segregated: In such a structure, each agent has access to only one task component. This structure is illustrated in Figure 6.8.

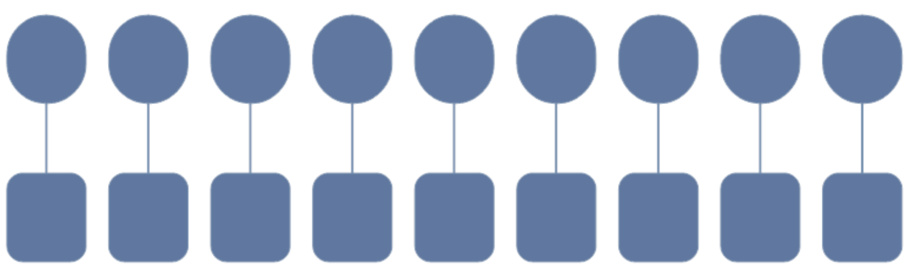

Figure 6.8: Segregated Resource Access Structure [Source: Lin and Carley [146]]

- Overlapped: In this structure, each agent has access to two task components, while each task component is accessible by only two analysts. Overlapped resource access structure is shown in Figure 6.9.

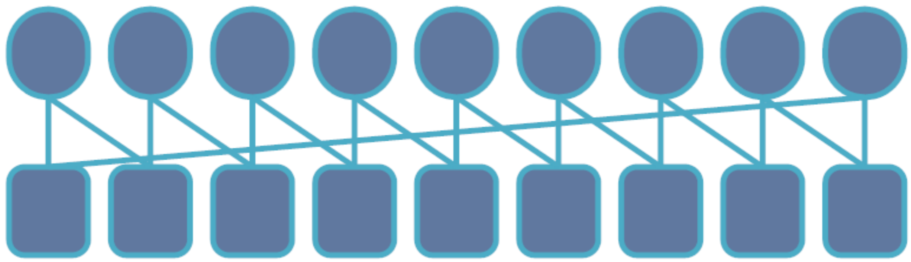

Figure 6.9: Overlapped Resource Access Structure [Source: Lin and Carley [146]]

- Blocked: In this type of structure, each agent has access to three task components. Three analysts have access to the exact same three task components. If these three analysts are in a hierarchical organisational structure, then they report to the same manager. This structure is illustrated in Figure 6.10.

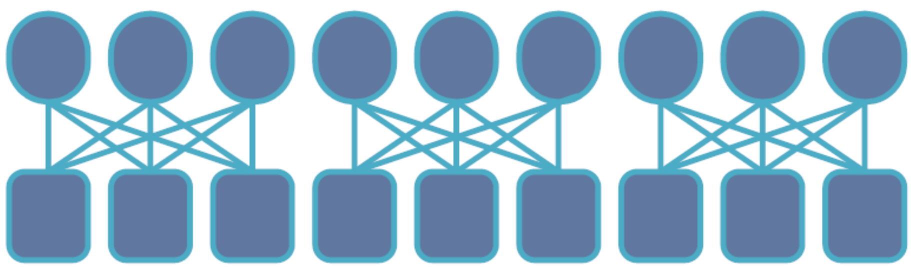

Figure 6.10: Blocked Resource Access Structure [Source: Lin and Carley [146]]

- Distributed: In this structure, each agent has access to three task components. No two analysts see the same set of task components. If these analysts are in a hierarchy or a matrix, then the manager would have indirect access to all the task components. Distributed resource access structure is shown in Figure 6.11.

By utilizing these structures, the radar task simulations are performed to determine the relative impact of cognition and design on organisational performance. If

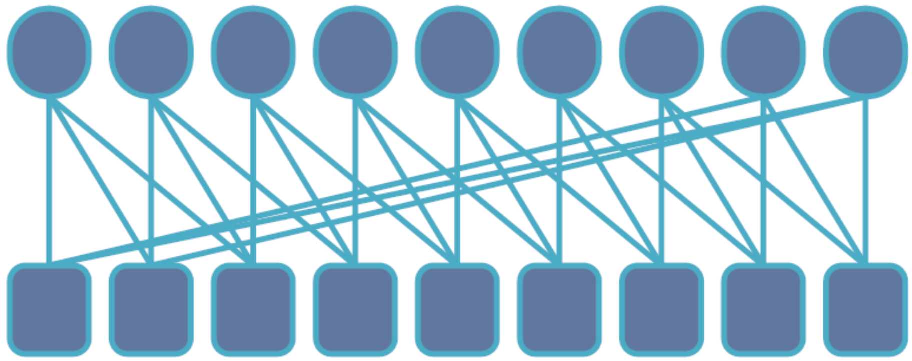

Figure 6.11: Distributed Resource Access Structure [Source: Lin and Carley [146]]

the true state of the aircraft is the same with the final decision of the organisation then the decision is correct. For a set of problems, the percentage of the correct decisions of those problems determines the performance of the organisation.

A number of radar task simulations are performed by adopting these organisational structures and resource access structures. In the following subsection the former radar task simulations are explained.

6.2.2 Former Radar Task Simulations

By implementing the radar task simulations, researchers attempt to analyse organisational performance. For this purpose, researchers use computational models, human experiments and archived data. These analyses can be performed at a micro (small group) and/or a macro (organisational) level. In the micro level studies, a hierarchy has a single tier. In other words, the hierarchy includes only one manager and 9 subordinates. In the macro level studies, a hierarchy may be multi-tier structure [48]. In the present study, only micro level studies are considered.

Carley et. al. [48] are the first researchers who studied the radar task simulations at the micro level. They adopted two resource access structures: the distributed structure and the blocked structure. At the same time, they adopted two organisational structures: team and hierarchy. They implemented the simulation in such a way that the organisations faced the same set of 30 tasks in the same order with the same organisational design while the agent models varied. In the first 30 tasks, the agents received feedback. For the second 30 tasks, the agents did not receive any feedback.

In their analysis, they performed a series of experiments which included computational models and humans. These are:

1. CORP-ELM,
2. CORP-P-ELM,
3. CORP-SOP,
4. Radar-SOAR, and
5. Human.

By using above artificial agents and humans, they aimed to compare their behaviour. These models vary in complexity and realism. CORP-SOP is the simplest model while the most complex agents are humans. Radar-SOAR agents are less complex than humans while they are more realistic than CORP models [48].

CORP is a computational framework which is a simulated testbed. This testbed is designed to enable the researchers to compare the performance of organisations with different settings. CORP models are artificial organisations consisting adaptive agents with task specific abilities [47].

The difference in CORP models is related to their decision making mechanism. CORP-SOP agents make decisions by following standard operating procedures provided by the organisation. CORP-ELM agents make decisions under the guidance of their own personal experience. CORP-P-ELM agents make decisions by guessing based on a probabilistic estimate of the obtained answers through their own experience [48].

SOAR agents are more complex artificial entities. They are based on the SOAR architecture which attempts to mimic general intelligence. Radar-SOAR is a composite simulation system, specifically designed for the radar task. It provides a way to compare and contrast the performance of SOAR agents on the radar task [49].

Carley et. al. [48] used four different organisational designs in their simulations:

1. A team with voting organisational structure and a blocked resource access,
2. A team with voting organisational structure and a distributed resource access,
3. A hierarchy with a single supervisor organisational structure and a blocked resource access, and
4. A hierarchy with a single supervisor organisational structure and a distributed resource access.

In all of these settings, there were nine subordinated (personnel) who obtain and analyse information. As stated before, in their experiment they used the same 60 tasks. The results of the radar task simulation performed by Carley et. al. [48] are shown in Table 6.2.

All of the results shown in this subsection are performance percentage. According to these results, the agent models performed better than humans in all team situations. Humans showed better performance than CORP-ELM and CORP-P-ELM in the hierarchy. In the distributed resource access structure humans performed better. However, CORP-P-ELM and Radar-SOAR performed better while the resource access is blocked. The other agent models, performed better in the distributed resource access like humans. CORP-P-ELM performed the worst in the hierarchy, while humans performed worst in the team with voting. The results indicate that the performance in a different organisational setting depends on the type of the employed agent.

Table 6.2: Radar Task Simulation Results of Carley et. al. [Source: [48]]

Agent	**Organisational Design**			
	Team		**Hierarchy**	
	Blocked	Distributed	Blocked	Distributed
CORP-ELM	88.3	85.0	45.0	50.0
CORP-P-ELM	78.3	71.7	40.0	36.7
CORP-SOP	81.7	85.0	81.7	85.0
Radar-SOAR	73.3	63.3	63.3	53.3
Human	50.0	56.7	46.7	55.0

Sun and Naveh [219] criticised these experiments stating that the agent models are being fairly simplistic. They stated that the intelligence level of these agents including SOAR was rather low. They added that learning was not complex enough to mimic human cognition. Based on these critiques they performed the same simulation by adopting a cognitive architecture called CLARION.

In their simulations, Sun and Naveh used the same organisational setting while employing the same number of agents. They replaced the agents with CLARION agents. Then they chose 100 tasks randomly. In these settings, they performed a number of simulations. The first simulation was a docking simulation in an abstract sense. They run the simulation for 4,000 cycles to match the human data. The results of this experiment and previous human data are shown in Table 6.3.

Table 6.3: Radar Task Simulation Results of Sun and Naveh [Source: [219]]

Agent	**Organisational Design**			
	Team		**Hierarchy**	
	Blocked	Distributed	Blocked	Distributed
Human	50.0	56.7	46.7	55.00
CLARION	53.2	59.3	45.0	49.4

All of the above results are performance percentage. According to these results, CLARION achieved the best performance match with human data.

In the second simulation they performed, they increased the number of cycles to 20,000. By this experiment, they proved that performance can be improved in the long run by the contribution of the learning approach adopted in CLARION. They criticised the original simulations being the result of limited training.

Their findings indicated that a team organisation using distributed access achieves a high level of performance quickly then the learning process slows down and the performance does not increase much. Contrary to this, a team with blocked access starts out slowly. However, it reaches the distributed access' performance in the long run. In the hierarchy, they stressed that learning is slower and more erratic; since, two layers of agents are being trained. They indicated that when a hierarchy in a blocked access, the performance is worse; since, there is very little learning.

In their third simulation, they varied a number of cognitive parameters and observed their effect on the performance. With this simulation, they confirmed the effects of the organisation structure and the resource access structure. Besides, they found that the interaction of the organisation structure and the resource access structure with the length of training was significant. In addition, they found no significant interaction between the learning rates with the organisational settings.

In their last two simulations, they introduced individual differences in the agents. Firstly they replaced one of the CLARION agents with a weaker agent. Then they performed simulation in a hierarchy with distributed resource access. Under these settings, the performance of the organisation dropped by only three to four percent. They concluded that the hierarchies are flexible enough to deal with a single weak performer. Secondly, they employed CLARION agents with a different learning rate. They found that the hierarchy performed better than team with voting. They claimed that supervisors could take individual differences into account by learning from experiences.

6.2.3 ReCau: The Radar Task Simulation

In the literature, the radar task is chosen to analyse interaction between design and cognition for several reasons. The most important reasons are that the radar task is inspired from a real world problem and widely examined. The other reason is that the task is a specific and well defined task. Thirdly, the true decision can be known and feedback can be provided. Fourth, multiple agents can be employed in a distributed environment so that the agents can work on different aspects of the task. Fifth, the task has a limited number of cases; therefore, mathematical techniques can be used to evaluate agent performance. The last but not the least important reason is that the task can be expanded further by including other factors.

Beside these reasons, in the present study the radar task is chosen in order to evaluate the decision-making mechanism of ReCau. The decision-making mechanism is the most significant component of the architecture. In the present study, several simulations are performed to test the decision-making mechanism. These simulations are implemented by using Java programming language.

In this dissertation study, the same organisational settings with the former studies are implemented. These settings are as follows:

1. Setting 1: Team with voting organisational structure and blocked resource access

2. Setting 2: Team with voting organisational structure and distributed resource access

3. Setting 3: Hierarchy with a single manager organisational structure and blocked resource access

4. Setting 4: Hierarchy with a single manager organisational structure and distributed resource access

In these settings, the same tasks are performed by ReCau agents. The most important difference between previous studies and this study is that the former studies chose a set of problems. They performed simulations for the chosen set of problems. In the present study, all of the tasks are generated randomly to realise a real unbiased environment. To achieve this aim, the features of each aircraft are generated randomly and independently from each other. The other difference is that in the present study the length of the simulation is higher.

Initially, a docking simulation is performed while the organisational structure is the team with voting while the resource access structure is blocked. By performing a docking simulation, cognitive parameters are adjusted to match human data. Then the same cognitive parameters are used in the other three settings.

Table 6.4: The Docking Simulation Results

Replication	Team		Hierarchy	
	Blocked	**Distributed**	**Blocked**	**Distributed**
1	52.45	53.68	42.345	42.63
2	53.49	53.67	42.995	43.02
3	52.935	54.03	42.345	42.7
4	52.765	53.515	42.545	43.07
5	52.61	53.915	42.535	42.615
6	53.17	53.16	42.665	42.63
7	53.165	53.245	42.17	43.86
8	53.365	53.81	42.225	42.97
9	52.995	53.545	42.82	42.64
10	53.61	53.325	43.19	42.715
Average Performance	53.0555	53.5895	42.5835	42.885
Mean Values	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75
Variance Value	0.085	0.085	0.085	0.085

The length (cycle) of the simulation is set to $20,000$ and then the simulation is run for 10 times for each setting separately. The results of the docking simulation are shown in Table 6.4. The results shown in the table are performance percentage.

Table 6.5: The Simulation Results when Variance is Reduced

Variable	Organisational Design			
	Team		Hierarchy	
	Blocked	Distributed	Blocked	Distributed
Average Performance	59.2075	61.04	49.679	50.879
Mean Values	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75
Variance Value	0.05	0.05	0.05	0.05
Average Performance	75.5565	72.2895	72.3165	76.5915
Mean Values	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75	0.65, 0.70, 0.75
Variance Value	0.01	0.01	0.01	0.01

The cognitive parameters of ReCau provide very high flexibility. To illustrate this aspect of ReCau, in the following experiments the variance value in the decision-making mechanism is reduced to low levels (0.05 and 0.01). The same length, number of replications, settings and mean values are used in these simulations. The results of these simulations are shown in Table 6.5. In this table, only the overall average performance percentages are shown.

As it can be seen in the table, while the variance value is reduced, the performance of the organisation is increasing. The highest performance is achieved in a hierarchy with distributed resource access.

In the following simulation, the difference between the mean values is increased. The results are shown in Table 6.6. In these simulations, the variance value is once again set to 0.085.

Table 6.6: The Simulation Results when Difference between Mean Values is Increased

Variable	**Organisational Design**			
	Team		**Hierarchy**	
	Blocked	Distributed	Blocked	Distributed
Average Performance	61.5195	63.5035	52.77	54.356
Mean Values	0.60, 0.70, 0.80	0.60, 0.70, 0.80	0.60, 0.70, 0.80	0.60, 0.70, 0.80
Variance Value	0.085	0.085	0.085	0.085
Average Performance	70.263	70.5555	64.2985	68.7015
Mean Values	0.50, 0.70, 0.90	0.50, 0.70, 0.90	0.50, 0.70, 0.90	0.50, 0.70, 0.90
Variance Value	0.085	0.085	0.085	0.085

As it can be seen, while the variance value is constant, if the difference between mean values is increased, the performance increases. However, in the previous simulation, better performance is observed. The detailed evaluation of the radar task simulation can be found in the following subsection.

6.2.4 The Evaluation of the Radar Task Simulation

In this dissertation study, the radar task simulation is undertaken to illustrate and evaluate the decision-making mechanism of ReCau. For this purpose, the radar task simulations are performed by varying cognitive parameters in the decision-making mechanism of ReCau.

In the Table 6.7, the simulation results of the existing architectures and the docking simulation results of ReCau are shown together to compare the results.

Table 6.7: Comparison of Docking Simulation Results

Agent	**Organisational Design**			
	Team		**Hierarchy**	
	Blocked	Distributed	Blocked	Distributed
CORP-ELM	88.3	85.00	45.00	50.00
CORP-P-ELM	78.3	71.7	40.0	36.7
CORP-SOP	81.7	85.0	81.7	85.0
Radar-SOAR	73.3	63.3	63.3	53.3
CLARION	53.2	59.3	45.0	49.4
ReCau	53.1	53.6	42.6	42.9
Human	50.0	56.7	46.7	55.0

Even though the same settings are adopted in all of these simulations, it must be noted that there are significant differences between these simulations. Above table is given to show the performance of all studies together to show the relative performance of ReCau.

As it can be seen in the table, the performance of the ReCau agents matches the human data well. A better match in this task means a closer performance percentage to human data. ReCau performance best matches human data in a team with voting organisational structure. It is because of the fact that the docking simulation is performed in this setting to determine cognitive parameters which can match human data. Then those cognitive parameters are used in the other three settings.

The performance of the CLARION agents also matches human data well. Especially, in the first three setting, the performance percentage difference between CLARION, ReCau and humans is around 3 percent. However, the performance of CLARION agents matches human data better in hierarchy with distributed resource access structure.

The performance pattern of ReCau agents also matches human data. As it can be seen, in the distributed resource access structures the ReCau performs slightly better. It is the same for human data. Humans also perform better in the distributed resource access structures. Humans and the ReCau agents show the highest performance in a team with distributed resource access structure. They show the lowest performance in a hierarchy with blocked resource access structure.

From these results, it can be deduced that the distributed resource access structure has positive impact over performance. Except for CORP-SOP agents, the performance of all agents is higher in a team. Therefore, it can be asserted that the agents including the ReCau agents perform better in a team.

In ReCau simulations, the learning approach proposed by ReCau is not implemented; since, social learning is not an appropriate approach to adopt in this type of a task. Under this setting, the findings support the results of Sun and Naveh [219]. They stated that very little learning takes place in a hierarchy with blocked access. The results of ReCau simulations confirm this finding; since, the performance of ReCau agents matches human data well without employing any learning approach.

The most significant performance difference between human data and the ReCau agents is observed in hierarchy with distributed resource access structure. The performance of the CLARION agents matches better in this setting. These results indicate that the learning is more effective in a hierarchy with distributed resource access structure.

Carley et. al. [48] state that at the micro levels, the same predictive performance accuracy with human data can be achieved by more cognitively accurate models. In the light of this fact, the findings of the docking simulation of ReCau indicate that the decision-making mechanism proposed along with ReCau is highly realistic.

When the cognitive parameters in the decision-making mechanism of ReCau are varied, higher performance percentages are achieved. Even though, the performance percentages of ReCau cannot go as high as CORP models, it still holds promise to perform like humans in this type of choice tasks.

The simulation results of ReCau also reveal interesting results regarding to the organisational theory. The performance percentages of ReCau agents confirm the results of Carley et. al. [48] who stated that the agent cognition interacting with organisational design affects organisational performance.

The results indicate that the organisational structure has more significant effect on performance than the resource access structure. In the docking simulation, the performance of ReCau agents is significantly different in different organisational structures. However, there is no significant performance difference while the agents are in the same organisational structure and the resource access structure is different.

Chapter 7

Discussion and Conclusion

Intelligent agent technologies are central to Artificial Intelligence research to simulate intelligent behaviour on computers. In the literature, most commonly approved attributes of intelligent agents are situatedness, autonomy and flexibility with social capabilities. Agent and cognitive architectures provide the means to satisfy those attributes to develop agents.

One of the major objectives in intelligent agent technologies is to develop believable agents. In order to be believable an agent should not only be situated, autonomous and flexible but also should be capable of displaying affect. In addition, believable agents should have a stronger sense of autonomy. To achieve stronger autonomy, agent architectures should enable an agent to be driven by their motives and learn from their experiences.

Believable agents should also have a realistic decision-making mechanism. The actions performed by the agents are determined through the decision-making process. The performance of the decision-making mechanism also has significant effect over the success of believable agents. Therefore, believable agents should have realistic decision-making mechanisms. Along with these properties, believable agents are essential to develop real-life like simulated environments.

7.1 The Summary of The Contributions

The intentional notion, on which the theories of agents are based, provides a good infrastructure. However, the intentional notion cannot explain the emergence of intelligent behaviour. Moreover, due to the rationality assumption of the intentional notion, some behaviours of intelligent beings cannot be explained by the intentional notion.

To overcome these shortcomings, by this research study the intentional notion and the theories of needs are merged. In this manner, it is attempted to explain the behaviours of intelligent entities by the combination of these two approaches. Instead of the rationality assumption, it is claimed that the most basic assumption of intelligence simulation is causality. In this context, intelligent entities observe their internal state and the environment. The observations can be considered as the cause, while the actions performed based on those observations can be considered as the effect. In this process, the needs are the nexus which provides a metric measure to select among alternative plans.

While addressing these issues, it is attempted to develop a general approach to simulate intelligent behaviour. According to the proposed approach, the term intelligence is defined as an abstract notion to express the cognitive processes of autonomous, situated, flexible, and social entities which can display affect and learn while they perform activities intentionally that are motivated by their needs.

By extending the theory of needs, in the proposed approach, emotions are attempted to be explained as a result of the satisfaction or dissatisfaction of the needs. Accordingly, a sufficient satisfaction of needs results in positive emotions while an insufficient satisfaction of needs results in negative emotions. Furthermore, in the proposed approach, if a need is strongly satisfied or dissatisfied, it results in stronger emotions. Stronger emotions can change the order of needs in the hierarchy. In this respect, the order of the needs is not fixed. Except for the existence needs, the order of needs can change. While a strong positive emotion moves the corresponding need downwards in the hierarchy, a strong negative emotion moves the associated need upwards in the hierarchy.

Even though this approach does not explain all aspects of effects of emotions on intelligence, it still explains a part of it. If a need moves up in the hierarchy, it means that an agent is going to avoid the conditions associated with that need. If a need moves down, it is more likely that the agent is going to pursue that need more frequently; since, in some sense the need gains priority. This means that the agent tends to pursue more favourable situations.

In addition, the proposed approach puts social learning theory into practice. Social learning theory explains how people learn from each other. This theory is adopted in order to provide a higher sense of autonomy to agents. While adopting social learning theory, reinforcement learning is used. Reinforcement is provided by the expected satisfaction which is consistent with the theory of needs.

In accordance with the proposed approach, an agent architecture is proposed. The architecture is called Reactive-Causal Architecture (ReCau). ReCau is meant to be a general purpose architecture that can be employed to develop believable intelligent agents to simulate either human or animal intelligence. To the best of our knowledge, ReCau is the first architecture incorporating all of the core attributes of believable agents together.

7.2 An Overall Evaluation of ReCau

In general, a ReCau agent is driven by pre-determined needs which provide a higher level of autonomy. ReCau includes a filtering mechanism which enables the planner to focus on only meeting one need at a time. The filtering mechanism also provides the means for resource boundedness by eliminating irrelevant data. By the motivation activator, a ReCau agent is enabled to focus on its lower level needs first.

The proposed approach adopts ERG theory in the light of the theory of needs. Like it is proposed in the original study, lower level needs are attempted to be satisfied first by ReCau agents. By following ERG, while developing ReCau agents, the order of the needs of each agent can be designed to be different. According to the ERG theory different levels of needs can be pursued simultaneously. However, in the proposed architecture, the needs cannot be pursued simultaneously. This may be considered in future work.

In ReCau, some other issues like emotion generation are purely based on Maslow's ideas. The emotion generation structure of the approach is inspired from Maslow who stated that if the lowest level needs are not satisfied, people would feel negative primitive emotions.

Another important issue is that while developing a ReCau agent, it is not necessary to incorporate all of the needs defined by Maslow. According to its design purposes, one can choose some of those needs and implement them accordingly. However, if one adopts the whole of Maslow's hierarchy, a ReCau agent would have more human-like intelligence.

The performance obstacles of the ReCau are related with the deliberative layer. The deliberative layer of ReCau contains slow components like a planner and a learner. In particular, the planner in ReCau is a discrete feasible planner; therefore, it may slow down the performance of ReCau agents. However, the ReCau architecture is put forward to realise the ideas presented in the proposed approach. Therefore, the architecture is not meant to be an efficient one at this stage.

When considering radar task simulation undertaken by ReCau agents, it can be said that the performance of the ReCau agents matches human data well. Especially, in the first three settings, the performance percentage difference between ReCau agents and humans is around 3 per cent. These similarities between humans and ReCau agents imply that the original simulations are the result of limited training. It is due to the fact that no learning approach is adopted by ReCau agents. However, the performance of ReCau agents do not match human data well in a hierarchy with distributed resource access structure. It can be said that learning is effective in the last setting. Additionally, the performance pattern of ReCau agents also matches human data; therefore, it can be stated that ReCau provides highly realistic decision-making mechanism.

In a recent study, Sun [217] presented his opinion on essential motivational representations necessary for a comprehensive computational cognitive architecture. In particular, he presented important criteria that agents must meet. These criteria are sustainability, purposefulness, focus and adaptivity. These criteria are explained below:

- Sustainability: An agent must be capable of satisfying its existence needs like hunger and thirst.
- Purposefulness: An agent must choose its actions in accordance with some criteria which should enhance the sustainability of the agent.
- Focus: An agent must be able to focus its activities. It should be in accordance with its purposes. In addition, the agent should be able to stop pursuing its activities, temporarily or permanently, or whenever a more urgent need arises.
- Adaptivity: An agent must be able to adapt its behaviour through learning. It should enable the agent to improve its purposefulness, sustainability, and focus.

During the implementation phase of ReCau, if all of the needs listed by Maslow are adapted, ReCau agents are capable of satisfying their existence needs. ReCau agents choose among alternative plans in accordance with the expected satisfaction obtained by undertaking the corresponding actions. Therefore, the satisfaction of needs provides a criterion to choose their actions. The filtering mechanism of ReCau, enables ReCau agents to focus only on activities which are relevant to their needs. The needs of ReCau agents are ordered in a hierarchy which provides the means to direct its focus on more urgent needs. Finally, ReCau agents are capable of learning socially which allows them to adapt their behaviours in the future actions. Therefore, ReCau agents are capable of meeting the criteria proposed by Sun.

Last but not the least considerable issue related with existing agent and cognitive architectures is that they employ a probabilistic approach to provide a means to select amongst alternative plans. But this approach lacks in explaining the effect of the motives on the decision-making process. This is because satisfaction is a metric measure in this process, not a probabilistic event.

In ReCau, satisfaction degrees are normally distributed random numbers coming from certain mean and variance values. As a result, the proposed approach provides a degree of randomness in the process which in turn explains human intelligent behaviour. Therefore, the decision-making mechanism of ReCau simulates the human behaviour better.

7.3 Further Research Directions

In its current form, Reactive-Causal Architecture provides a good infrastructure for believable agents. However, in the future a number of improvements can be made.

In the future, hierarchical and/or the non-linear planning approaches can be adopted to increase the efficiency of ReCau. In its current form, search is highly complex. A hierarchical planning approach can help reducing the complexity of the search. Non-linear planning approaches can provide a partial ordering of the plans. In this manner, search time can be reduced.

The type of learning adapted in ReCau is based on the theories of social learning. In addition to this type of learning, an approach to enable the ReCau agents to learn from their mistakes can be incorporated. For this purpose, Q-learning can be adapted in the near future.

Adapting the Q-learning approach in ReCau can facilitate the development and implementation of applications. After implementing the Q-learning approach, the radar task simulation may be undertaken again. In this manner, a better explanation on the effect of learning on organisational design can be obtained.

In the ReCau architecture, second central moment (variance) is used. Instead of variance, standard deviation might have been used; since, the variance is the positive square root of standard deviation. There are also higher central moments like skewness and kurtosis. Skewness is a measure of the lopsidedness of the distribution. If the tail of the distribution is heavier on the left, then it can be said that distribution is skewed to the left. This type of skewness is called negative skewness. A distribution skewed to the right have positive skewness. The fourth central moment called kurtosis is a measure of whether the distribution is tall and skinny or short and squat [65].

In the reviewed literature, there is no cognitive agent architecture that adopts higher order moments in the decision-making mechanism. In the long term, higher central moments can be considered to be adopted in the decision-making mechanism of ReCau. In particular, it is possible that human satisfaction on some plan alternatives might be skewed to the left or right. If that is the case, it may be worthwhile to adopt higher central moments in ReCau.

In the last decade, there is a growing interest in uncertainty. Researchers state that it is important to incorporate a model to represent and reason under uncertainty to apply agents in real world domains. For this purpose, there are some models like Graded BDI which extends Belief, Desire, Intention approach [50]. In the future, ReCau can be extended further to work under uncertainty by adopting approaches like Graded BDI.

The potential applications of ReCau in domains such as social simulation, embedded systems and entertainment computing can be envisioned. In the future, implementations of ReCau in these domains can be pursued.

In particular, ReCau with its realistic decision-making mechanism provides a good model for agent-based simulations. By employing ReCau, several agent-based social simulations can be performed. As in the radar task, agent-based social simulations can help understanding the underlying organisational theory.

After further increasing the efficiency of planning approach adopted in ReCau, it can be employed in embedded systems. ReCau can be employed to perform a few dedicated functions with real-time computing constraints. It must be noted that, to meet real-time computing constraints, the efficiency of the discrete feasible planner must be improved. Since ReCau achieves high levels of performance in the radar task, it can also be used in applications which require human expertise such as air traffic control systems. In an air traffic control system, the radars include one or more embedded systems of their own. ReCau can be adopted to be employed in such radars.

ReCau with its social learning capability can also be employed in entertainment computing applications. Recently, massively multiplayer online games like World of Warcraft have become very popular in the leisure industry. Such games include real players interacting with each other and non-player characters. In order to be a realistic application, these games require non-player characters that are capable of showing human-like intelligence. These types of characters can be realised by adopting ReCau.

The social learning capability of ReCau can help non-player characters to adopt behaviours of the real players. Especially, this capability can enhance the realism of first person shooter games. In the games like Quake, Half Life and Counter Strike, no matter how hard the game setting is, human players can develop strategies to overcome non-player characters. Social learning can help non-human characters to learn the strategies developed by humans.

7.4 Concluding Remarks

In the ReCau architecture, ERG theory is adopted to explain motives. By following ERG, needs of the each agent can be ordered differently. This enables agents to have different personalities somehow. Moreover, a ReCau agent is also capable of affect display. Under the light of the Maslow's explanations, emotions are explained as a result of the satisfaction or the dissatisfaction of needs.

In the contemporary literature, many researchers have studied the effects of emotions on the decision-making process. The proposed- approach extends the theories of needs to explain this aspect of intelligence. It is proposed that stronger emotions can change the order of the associated needs. If a particular need is strongly satisfied or dissatisfied, it results in a strong emotion. While a strong positive emotion moves associated need downwards in the hierarchy, a strong negative emotion moves associated need upwards.

The causal layer works in accordance with the causality principle. The decision-making and the emotion generation mechanisms operate under the cause and effect concept. The decision-making and the emotion generation mechanisms use needs as a nexus which provides the means to select among alternatives. Instead of probabilities, the causal layer employs random number generation mechanisms to provide a measure for selecting among alternatives.

By providing a degree of randomness in the decision-making process of the architecture, ReCau holds promise to simulate human intelligent behaviour better; since, it provides an enhanced action flexibility. The approach employed in ReCau provides flexibility in such a way that the behaviours of agents cannot be predicted while supporting routine activities.

Apart from this merit of the decision-making mechanism of ReCau, it also supports the development of believable agents. The radar task simulation results indicate that ReCau provides a very good human cognition model. To develop realistic believable agents, one of the most important features is a realistic decision-making mechanism; since, it determines the actions taken.

By varying the cognitive parameters in the decision-making mechanism, ReCau agents can achieve high performance in the radar task. In the future, by adopting an appropriate learning approach to learn from mistakes, ReCau agents can achieve better performance.

In conclusion, ReCau is meant to be general purpose architecture. It is the first intelligent agent architecture incorporating strong autonomy, flexibility, situatedness and affect display attributes all together. ReCau can be employed to develop believable intelligent agents with its highly realistic decision-making mechanism in many real-world domains of interest.

Appendix A

Glosarry

3APL	An Abstract Agent Programming Language
AAS	Autonomous Agent Systems
ABCL	Actor-Based Concurrent Language
ACL	Agent Communication Languages
ACT-R	Adaptive Control of Thought Rational
AI	Artificial Intelligence
AIS	Adaptive Intelligent System
ANA	Agent Network Architecture
AOR	Agent-Object-Relationship
A-Team	Asynchronous Team
AUTODRIVE	Agens in a Simulated Driving World
BICA	Brain Inspired Cognitive Architecture
BDI	Blief, Desire, Intention
CBR	Case-Based Reasoning
CHREST	CHunk Hierarchy and REtrieval Structures

CLARION	Connectionist Learning with Adaptive Rule Induction ON-line
CMattie	Conscious Mattie
CogAff	Cognition and Affect
CORP	A Computational Framework
DAI	Distributed Artificial Intelligence
DARPA	Defence Advanced Research Projects Agency
DAS	Distributed Asynchronous Scheduler
dMARS	Distributed Multi-Agent Reasoning System
EPAM	Elementary Perceiver and Memorizer
EPIC	Executive-Process/Interactive Controls
ERE	Entropy Reduction Engine
ERG	Existence, Relatedness and Growth
FA/C	Functionally Accurate Model
FIPA	Foundation for Intelligent Physical Agents
FORR	FOr the Right Reasons
GLAIR	Grounded Layered Architecture with Integrated Reasoning
GWT	Global Workspace Theory
H-CogAff	An Instance of CogAff Architecture Scheme
IA	Intelligent Agent
IPA	Foundation for Intelligent Physical Agents
IRMA	Intelligent Resource-Bounded Machine Architecture
JADE	Java Agent Development Framework
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LISP	LISt Processing

LEABRA	Local, Error-driven and Associative, Biologically Realistic Algorithm
LRMB	A Layered Reference Model of the Brain
MA	Motivated Agency
MAS	Multi-Agent System
MAS-SOC	Multi-Agent Simulation for the SOCial sciences
MEA	Means-Ends Analysis
MPA	Motive Processing Architecture
NMRAA	New Millennium Remote Agent Architecture
PDDL	Planning Domain Definition Language
pdf	Probability Density Function
PoB	Patterns of Behaviour
PRS	Procedural Reasoning System
RAA	Rational Agent Architecture
ReCau	Reactive-Causal Architecture
RCS	Real-Time Control System
SAC	Source of Activation Confusion
SAL	Synthesis of ACT-R and Leabra
SOAR	State, Operator and Result
SRI	Stanford Research Institute
STRIPS	STanford Research Institute Problem Solver
SUO-KIF	Standard Upper Ontology Knowledge Interchange Format

Bibliography

[1] A. D. Channon abd R. I. Damper. The Evolutionary Emergence of Socially Intelligent Agents. In *Socially Situated Intelligence*, pages 41–49, Zurich, 1998.

[2] G. Abdelkader. Requirements for Achieving Software Agents Autonomy and Defining their Responsibility. In *The Autonomy Workshop at Autonomous Agents and Multi Agent Systems*, Melbourne, 2003.

[3] Alfred Adler. *A Systematic Presentation in Selections from Alfred Adler Writings*, chapter Individual Psychology of Alfred Adler. HarperCollins, 1989.

[4] Philip E. Agre. *The Dynamic Structure of Everyday Life*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.

[5] Philip E. Agre and David Chapman. PENGI: An Implementation of a Theory of Activity. In *The Sixth National Conference on Artificial Intelligence*, pages 268–272. Seattle, Washington, 1987.

[6] James S. Albus. RCS: A reference model architecture for intelligent control. *Computer*, 25(5):56–79, 1992.

[7] Clayton P. Alderfer. *Existence, Relatedness, and Growth: Human Needs in Organizational Settings*. Free Press, New York, 1972.

[8] Jose A. Ambros-Ingerson and Sam Steel. Integrating Planning, Execution and Monitoring. In *The Seventh National Conference on Artificial Intelligence*, pages 83–88, St. Paul, Minnesota, 1988.

[9] John R. Anderson. *Language, Memory, and Thought*. Erlbaum, Hillsdale, New Jersey, 1976.

[10] John R. Anderson. *The Architecture of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1983.

[11] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.

[12] John R. Anderson and Christian J. Lebiere. *The Atomic Components of Thought*. Erlbaum, Mahwah, New Jersey, 1998.

[13] Rita L. Atkinson, Richard C. Atkinson, and Ernest R. Hilgard. *Introduction to Psychology*. Harcourt Brace Jovanovich, San Diego, eighth edition, 1987.

[14] John Langshaw Austin. *How to Do Things with Words*. Oxford University Press, Oxford, first edition, 1962.

[15] Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, 1988.

[16] Bernard J. Baars. *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press, 1997.

[17] Bernard J. Baars. The conscious access hypothesis: Origins and recent evidence. *Trends in Cognitive Sciences*, 6(1):47–52, 2002.

[18] Penny Baillie and Dickson Lukose. An affective decision making agent architecture using emotion appraisals. *Lecture Notes in Computer Science*, 2417:581–590, 2002.

[19] T. C. Adalet Bakanliği. Türk Ceza Kanunu, 2005.

[20] Albert Bandura. *Principles of Behavior Modification*. Holt, Rinehart and Winston, New York, 1969.

[21] Albert Bandura. *Social Foundations of Thought and Action*. Prentice-Hall, 1985.

[22] Joseph Bates. The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125, 1994.

[23] Joseph Bates, A. Bryan Loyall, and W. Scott Reilly. An Architecture for Action, Emotion, and Social Behavior. In *The Fourth European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pages 13–26, S.Martino, 1992.

[24] Luc P. Beaudoin. *Goal Processing in Autonomous Agents*. PhD thesis, University of Birmingham, Birmingham, 1994.

[25] Luc P. Beaudoin and Aaron Sloman. A Study of Motive Processing and Attention. In Aaron Sloman, D. Hogg, G. Humphreys, D. Partridge, and A. Ramsay, editors, *The Prospects for Artificial Intelligence*, page 229178. IOS Press, 1993.

[26] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE - A FIPA-Compliant Agent Framework. Technical report, CSELT, 1999.

[27] Guido Boella and Rossana Damiano. An architecture for normative reactive agents. *Lecture Notes in Artificial Intelligence*, 2413:1–17, 2002.

[28] R. Peter Bonasso. Integrating Reaction Plans and Layered Competences Through Synchronous Control. In *The International Joint Conference on Artificial Intelligence*, pages 1225–1233, 1991.

[29] R. Peter Bonasso and David Kortenkamp. Using a Robot Control Architecture to Automate Space Shuttle Operations. In *The 1997 National Conference on Artificial Intelligence*, pages 949–956. AAAI Press, 1997.

[30] Rafael H. Bordini, Jomi F. Hlibner, and Renata Vieira. *Multi-Agent Programming*, chapter Jason and the Golden Fleece of Agent-Oriented Programming. Springer, 2006.

[31] Rafael H. Bordini, Fabio Y. Okuyama, Denise de Oliveira, Guilherme Drehmer, and Romulo C. Krafta. The mas-soc approach to multi-agent based simulation. *Regulated Agent-Based Social Systems*, 2934:70–91, 2004.

[32] George Edward Pelham Box and Mervin Edgar Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.

[33] Stevo Bozinovski and Liljana Bozinovska. Self-learning agents: A connectionist theory of emotion based on crossbar value judgment. *Cybernetics and Systems*, 32(6):637–669, 2001.

[34] Jeffrey M. Bradshaw. *Software Agent*. MIT Press, Cambridge, Massachusetts, 1997.

[35] Michael E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, Massachusetts, 1987.

[36] Michael E. Bratman, David Israel, and Martha Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:249–355, 1988.

[37] Franz Brentano. *Psychology from an Empirical Standpoint.* Routledge and Kegan Paul, London, 1874.

[38] Chris R. Brewin. Cognitive change processes in psychotherapy. *Psychological Review*, 96(3):379–394, 1989.

[39] Rodney Allen Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[40] Rodney Allen Brooks. *Designing Autonomous Agents*, chapter Elephants don't Play Chess, pages 3–15. MIT Press, 1990.

[41] Rodney Allen Brooks. Intelligence without Reason. In *The Twelfth International Joint Conference on Artificial Intelligence*, Sydney, 1991.

[42] Rodney Allen Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[43] Birgit Burmeister and Kurt Sundermeyer. Cooperative Problem Solving Guided by Intentions and Perception. In *The Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds*, pages 77–92, Amsterdam, 1992. Elsevier Science Publishers.

[44] Antonio Camurri and Alessandro Coglio. An architecture for emotional agents. *Multimedia*, 5(4):24–33, 1998.

[45] Walter Bradford Cannon. *The Wisdom of the Body.* Norton, New York, 1939.

[46] Cosmin Carabelea, Olivier Boissier, and Adina Florea. Autonomy in multi-agent systems: A classification attempt. *Lecture Notes in Computer Science*, 2969:103–113, 2004.

[47] Kathleen M. Carley and Zhiang Lin. Organizational designs suited to high performance under stress. *IEEE - Systems Man and Cybernetics*, 25(1):221–230, 1995.

[48] Kathleen M. Carley, Michael J. Prietula, and Zhiang Lin. Design versus cognition: The interaction of agent cognition and organizational design on organizational performance. *Journal of Artificial Societies and Social Simulation*, 1(3), 1998.

[49] Kathleen M. Carley and Mei Ye. Radar-soar: Towards an artificial organization composed of intelligent agents. *Journal of Mathematical Sociology*, 20:219–246, 1995.

[50] Ana Casali, Lluis Godo, and Carles Sierra. Graded bdi models for agent architectures. *Lecture Notes in Computer Science*, 3487, 2005.

[51] Nick L. Cassimatis. *Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.

[52] Krzysztof Cetnarowicz, Pablo Gruer, Vincent Hilaire, and Abder Koukam. A formal specification of m-agent architecture. *Lecture Notes in Computer Science*, 2296:62–72, 2001.

[53] Sanjay Chandrasekharan and Babak Esfandiari. Software Agents and Situatedness: Being Where? In *The the Proceedings of the Eleventh Mid-west Conference on Artificial Intelligence and Cognitive Science*, Menlo Park, California, 2000.

[54] Li Changhong, Li Minqiang, and Kou Jisong. Cooperation Structure of Multi-Agent and Algorithms. In *The 2002 IEEE International Conference on Artificial Intelligence Systems*, pages 58–65, 2002.

[55] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–378, 1987.

[56] David Chapman. *Instruction Use in Situated Activity.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990.

[57] Alexandra M. Coddington and Michael Luck. A motivation-based planning and execution framework. *International Journal on Artificial Intelligence Tools*, 13(1):5–25, 2004.

[58] Paul R. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32–48, 1989.

[59] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[60] Philip R. Cohen and Hector J. Levesque. *Intentions in Communication*, chapter Rational Interaction as the Basis for Communication, pages 221–256. MIT Press, Cambridge, Massachusetts, 1990.

[61] Philip R. Cohen and C. R. Perrault. Elements of a plan based theory of speech acts. *Cognitive Science*, 3:177–212, 1979.

[62] Jonathan H. Connell. SSS: A Hybrid Architecture Applied to Robot Navigation. In *The IEEE Conference on Robotics and Automation*, pages 2719–2724, 1992.

[63] Daniel D. Corkill. Blackboard systems. *AI Expert*, 6:40–47, 1991.

[64] John Woldemar Cowan. *The Complete Lojban Language.* Logical Language Group Inc., 1997.

[65] Michael J. Crawley. *The R Book.* Wiley, 2007.

[66] Daniel Crevier. *AI: Tumultuous History of the Search for Artificial Intelligence.* Basic Books, New York, 1993.

[67] Antonio R. Damasio. *Descartes' Error, Emotion Reason and the Human Brain.* Grosset/Putnam Books, New York, 1994.

[68] Mehdi Dastani, M. Birna van Riemsdijk, Frank Dignum, M. Birna, Riemsdijk Frank Dignum, and John-Jules Ch. Meyer. A Programming Language for Cognitive Agents Goal Directed 3APL. In *The First Workshop on Programming Multiagent Systems: Languages, Frameworks, Techniques, and Tools*, pages 111–130, 2003.

[69] Daniel Clement Dennett. Intentional systems. *The Journal of Philosophy*, 68(4):87–106, 1971.

[70] Daniel Clement Dennett. *The Intentional Stance.* MIT Press, Cambridge, Massachusetts, 1987.

[71] Daniel Clement Dennett. Three Kinds of Intentional Psychology. In Robert J. Stainton, editor, *The Perspectives in the Philosophy of Language: A Concise Anthology*, pages 163–186. Broadview Press, 2000.

[72] Oxford English Dictionary. Random, 2011.

[73] Mark D'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dmars. *Lecture Notes in Artificial Intelligence*, 1365:155–176, 1998.

[74] Mark D'Inverno, Michael Luck, Michael Georgeff, David Kinny, and Michael Wooldridge. The dMARS architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems*, 9(1):5–53, 2004.

[75] Mark D'Inverno, Michael Luck, and Michael Wooldridge. Cooperation Structures. In *The Fifteenth International Joint Conference on Artificial Intelligence*, pages 600–605, 1997.

[76] Mark Drummond, John Bresinat, and Smadar Kedar. The entropy reduction engine: Integrating planning, scheduling, and control. *ACM SIGART Bulletin*, 2(4):61–65, 1991.

[77] Clark D. Elliott. *The Affective Reasoner: A Process Model of Emotions in a Multi-Agent System*. PhD thesis, Northwestern University, Illinois, 1992.

[78] Britannica Concise Encyclopaedia. Motivation, 2005.

[79] Robert S. Engelmore and Anthony Morgan, editors. *Blackboard Systems*. Addison-Wesley, Reading, Massachusetts, 1988.

[80] Susan L. Epstein. For the right reasons: The FORR architecture for learning in a skill domain. *Cognitive Science*, 18(3):479–511, 1994.

[81] Edward A. Feigenbaum and Julian Feldman. *Computers and Thought*. McGraw Hill, New York, 1963.

[82] Edward A. Feigenbaum and Herbert A. Simon. EPAM-like models of recognition and learning. *Cognitive Science*, pages 305–336, 1984.

[83] Richard D. Fennell and Victor R. Lesser. Parallelism in artificial intelligence problem solving: A case study of hearsay ii. *IEEE Transactions on Computers: C*, 26(2):98–111, 1977.

[84] Innes Andrew Ferguson. *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, University of Cambridge, Clare Hall, 1992.

[85] Leon Festinger. *A Theory of Cognitive Dissonance*. Stanford University Press, 1957.

[86] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189–208, 1971.

[87] Maria Fox and Derek Long. PDDL2.1 : An extension to pddl for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.

[88] Michael Freed. Managing Multiple Tasks in Complex, Dynamic Environments. In *The Fifteenth National Conference on Artificial Intelligence*, pages 921–927, 1998.

[89] Sigmund Freud. *Beyond the Pleasure Principle*. Boni and Liveright Publishers, New York, 1922.

[90] Nguyen G., Dang T.T., Hluchy L., Laclavik M., Balogh Z., and Budinska I. Agent Platform Evaluation and Comparison. Technical report, Slovak Academy of Sciences, 2002.

[91] Erann Gat. *Reliable Goal-Directed Reactive Control for Real-world Autonomous Mobile Robots*. PhD thesis, Virginia Polytechnic Institute, Blacksburg, Virginia, 1991.

[92] Erann Gat. On three-layer architectures. *Artificial Intelligence and Mobile Robots*, pages 195–210, 1998.

[93] Russell G. Geen. *Human Motivation: A Social Psychological Approach*. Brooks/Cole Publication, 1995.

[94] Michael R. Genesereth and Richard E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical report, Computer Science Department, Stanford University, 1992.

[95] Michael P. Georgeff. Communication and Interaction in Multi-Agent Planning. In *The Proceedings of the Third National Conference on Artificial Intelligence*, pages 125–129, Washington, D.C., 1983.

[96] Michael P. Georgeff and Amy L. Lansky. Reactive Reasoning and Planning. In *The Sixth National Conference on Artificial Intelligence*, pages 677–682, Seattle, Washington, 1987.

[97] Georg Glaeser and Ernst Strouhal. Von Kempelen's Chess-Playing Pseudo-Automaton and Zu Racknitz's Explanation of its Controls 1979. In *The International Symposium on History of Machines*, pages 351–360. Kluwer Academic Publishers, 2000.

[98] Fernand Gobet, Peter C. R. Lane, Steve Croker, Peter C-H. Cheng, Gary Jones, Iain Oliver, and Julian M. Pine. Chunking mechanisms in human learning. *TRENDS in Cognitive Sciences*, 5:236–243, 2001.

[99] Daniel Goleman. *Emotional Intelligence: Why It Can Matter More than IQ*. Bloomsbury Publishing, London, 1996.

[100] Steve B. Goschnick. *Shadowboard: An Agent Architecture for Enabling a Sophisticated Digital Self*. PhD thesis, University of Melbourne, Melbourne, 2001.

[101] James L. Gould and Carol Grant Gould. *The Animal Mind*. Scientific American Library, New York, 2000.

[102] Jonathan Gratch. Èmile: Marshalling Passions in Training and Education. In *Agents*, pages 325–332, 2000.

[103] KQML Advisory Group, Tim Finnin, Don Mckay, and Rich Fritzson. An Overview of KQML: A Knowledge Query and Manipulation Language, 1992.

[104] V. Gyurjyan, D. Abbott, G. Heyes, E. Jastrzembski, C. Timmer, and E. Wolin. FIPA Agent Based Network Distributed Control System. In *The Computing in High Energy and Nuclear Physics*, La Jolla, California, 2003.

[105] Afsaneh Haddadi. A Hybrid Architecture for Multi-Agent Systems. In *The 1993 Workshop on Cooperating Knowledge Based Systems*, pages 13–26, 1994.

[106] Mehdi T. Harandi and G. Rendon. Classification and Organizational Issues in Distributed Problem Solving. In *The Proceedings of the 1998 ACM Symposium on Applied Computing*, pages 65–69, 1998.

[107] Shlomo Hareli and Brian Parkinson. What's social about social emotions? *Journal for the Theory of Social Behaviour*, 38(2):131–156, 2008.

[108] Paul Harmon, Rex Maus, and William Morrissey. *Expert Systems Tools and Applications*. John Wiley and Sons, New York, 1988.

[109] Inman Harvey, Ezequiel Di Paolo, Elio Tuci, Rachel Wood, and Matt Quinn. Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1-2):79–98, 2005.

[110] Nick Hawes. An Anytime Planning Agent For Computer Game Worlds. In *The Third International Conference on Computers and Games*, page 117, 2002.

[111] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, 1985.

[112] Barbara Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72:329–365, 1995.

[113] Henry Hexmoor, Johan Lammens, and Stuart C. Shapiro. Embodiment in GLAIR: A Grounded Layered Architecture with Integrated Reasoning for Autonomous Agents. In *The Florida AI Research Symposium*, page 325179, 1993.

[114] Douglas R. Hofstadter. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, 1995.

[115] Nick Howden, Ralph Rnnquist, Andrew Hodgson, and Andrew Lucas. JACK Intelligent Agents: Summary of an Agent Infrastructure. In *The Fifth International Conference on Autonomous Agents*, 2001.

[116] Marcus J Huber. JAM: A BDI-Theoretic Mobile Agent Architecture. In *The International Conference on Autonomous Agents*, pages 236–243. ACM Press, 1999.

[117] Ricardo Imbert and Angèlica de Antonio. When emotion does not mean loss of control. *Lecture Notes in Computer Science*, 3661:152–165, 2005.

[118] Nicholas R. Jennings. Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems Management*, 2(3):289–318, 1993.

[119] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:717, 1998.

[120] David J. Jilk, Christian Lebiere, Randall C. OReilly, and John R. Anderson. Sal: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 20(3):197–218, 2008.

[121] Richard A. Johnson and Gouri K Bhattacharyya. *Statistics: Principles and Methods*. John Wiley and Sons, Inc, 2009.

[122] Carl Jung. *Man and his Symbols*. Laurelleaf, New York, 1997.

[123] Leslie Pack Kaelbling. An Architecture for Intelligent Reactive Systems. In *Reasoning About Actions and Plans*, pages 395–410, San Mateo, California, 1986. Morgan Kaufmann Publishers.

[124] Hans Kamp. Discourse representation theory: What it is and where it ought to go. *Lecture Notes In Computer Science*, 320:84–111, 1988.

[125] Samin Karim, Liz Sonenberg, and Ah-Hwee Tan. A hybrid architecture combining reactive plan execution and reactive learning. *Lecture Notes in Artificial Intelligence*, 4099:200–211, 2006.

[126] Walter Kaufmann. *The Portable Nietzsche*. Viking Press, New York, 1967.

[127] Alan Kay. Computer software. *Scientific American*, 251(3):53–59, 1984.

[128] David E. Kieras and David E. Meyer. An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4):391–438, 1997.

[129] Philip Leroy Kilbride. *Plural Marriage for Our Times: A Reinvented Option?* Bergin and Garvey, 1994.

[130] Craig A. Knoblock. Learning Abstraction Hierarchies for Problem Solving. In *The Eighth National Conference on Artificial Intelligence*, pages 923–928, Boston, Massachusetts, 1990. AAAI Press.

[131] Samuel Noah Kramer. *Sumerian Mythology: A Study of Spiritual and Literary Achievement in the Third Millennium BC.* University of Pennsylvania Press, 1972.

[132] Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297175, 1995.

[133] John Laird. Extending the Soar Cognitive Architecture. In *The First Conference on Artificial General Intelligence*, 2008.

[134] John Laird, Allen Newell, and Paul S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[135] John E. Laird and Robert E. Wray III. Cognitive Architecture Requirements for Achieving Artificial General Intelligence. In *The Third Conference on Artificial General Intelligence*, 2010.

[136] John E. Laird, Robert E. Wray III, Robert P. Marinier, and Pat Langley. Claims and Challenges in Evaluating Human-Level Intelligent Systems. In *The Second Conference on Artificial General Intelligence*, 2009.

[137] Pat Langley, John E. Laird, and Seth Rogers. Cognitive Architectures: Research Issues and Challenges. Technical report, Institute for the Study of Learning and Expertise, Palo Alto, California, 2002.

[138] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10, 2009.

[139] Pat Langley, Kathleen B. McKusick, John A. Allen, Wayne F. Iba, and Kevin Thompson. A design for the ICARUS architecture. *ACM SIGART Bulletin*, 2(4):104–109, 1991.

[140] Steven Michael LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[141] Richard S. Lazarus. Thoughts on the relations between emotions and cognition. *American Physiologist*, 37(10):1019–1024, 1982.

[142] Jaeho Lee, Marcus J. Huber, Edmund H. Durfee, and Patrick G. Kenny. UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. In *The AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS)*, pages 842–849, 1994.

[143] Jimmy Ho-Man Lee and Lei Zhao. A real-time agent architecture: Design, implementation and evaluation. *Lecture Notes in Computer Science*, 2413:18–32, 2002.

[144] Victor R. Lesser. A Retrospective View of FA/C Distributed Problem Solving, 1991.

[145] Victor R. Lesser and Leed D. Erman. Distributed interpretation: A model and experiment. *IEEE Transactions on Computers: C*, 29(12):1144–1163, 1980.

[146] Zhiang Lin and Kathleen M. Carley. Maydays and Murphies - A Study of the Effect of Organizational Design Taks, and Stress on Organizational Performance. Technical report, Learning Research and Development Center, Pittsburgh University, 1992.

[147] Zhiang Lin and Kathleen M. Carley. DYCORP: A computational framework for examining organizational performance under dynamic conditions. *Journal of Mathematical Sociology*, 20(2-3):193–217, 1995.

[148] John W. Lloyd and Tim D. Sears. An architecture for rational agents. *Lecture Notes in Computer Science*, 3904:51–71, 2006.

[149] Edwin A. Locke and Gary P. Latham. Building a practically useful theory of goal setting and task motivation. *American Psychologist*, 57(9):705–717, 2002.

[150] Michael Luck and Mark D'Inverno. A Formal Framework for Agency and Autonomy. In *The First International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press / MIT Press, 1995.

[151] Màrcia Maçãs, Rodrigo Ventura, Luis Custódio, and Carlos A. Pinto-Ferreira. DARE: An Emotion-Based Agent Architecture. In *The FLAIRS-2001 Conference*, pages 150–154, 2001.

[152] Pattie Maes. The agent network architecture: ANA. *SIGART Bulletin*, 2(4):115–120, 1991.

[153] James B. Marshall. A self-watching model of analogy-making and perception. *Journal of Experimental and Theoretical Artificial Intelligence*, 18(3):267–307, 2006.

[154] James B. Marshall and Douglas R. Hofstadter. The metacat project: A self-watching model of analogy-making. *Cognitive Studies*, 4(4), 1997.

[155] Abraham Harold Maslow. *Motivation and Personality*. Harper and Row, 1987.

[156] David McAllester and David Rosenblitt. Systematic Nonlinear Planning. In *The Ninth National Conference on Artificial Intelligence*, pages 634–639, Anaheim, 1991.

[157] John McCarthy. *Ascribing Mental Qualities to Machines*. Morgan Kaufmann Publishers, Stanford, 1978.

[158] T. Lee McCauley and Stan Franklin. An Architecture for Emotion. In *The AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*, pages 122–127, 1998.

[159] David Clarence McClelland. *The Achieving Society*. Van Nostrand, Princeton, New Jersey, 1961.

[160] Pamela McCorduck. *Machines who Think*. Freeman, San Francisco, 1979.

[161] Robert C. Moore. A Formal Theory of Knowledge and Action. In J.F. Allen, J. Hendler, and A. Tate, editors, *The Readings in Planning*, pages 480–519, San Francisco, California, 1990. Morgan Kaufmann Publishers.

[162] Jörg P. Müller. A Conceptual Model for Agent Interaction. In *The Second International Working Conference on Cooperating Knowledge Based Systems*, pages 213–234, 1994.

[163] Jörg P. Müller, Markus Pischel, and Michael Thiel. Modelling Reactive Behaviour in Vertically Layered Agent Architectures. In Michael Wooldridge and Nicholas R. Jennings, editors, *The ECAI Workshop on Agent Theories, Architectures, and Languages*, pages 261–276. Springer-Verlag, 1995.

[164] Henry Alexander Murray. *Explorations in Personality: A Clinical and Experimental Study of Fifty Men of College Age*. Oxford University Press, 1970.

[165] Karen L. Myers. User Guide for the Procedural Reasoning System. Technical report, Artificial Intelligence Center SRI International, Menlo Park, California, 1997.

[166] Karen L. Myers and David E. Wilkins. The Act Formalism. Technical report, SRI International Artificial Intelligence Center, Menlo Park, California, 1997.

[167] Ulric Neisser. *Cognition and Reality: Principles and Implications of Cognitive Psychology*. W. H. Freeman, San Francisco, 1976.

[168] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, 1994.

[169] Allen Newell and Herbert A. Simon. *Computers and Thought*, chapter GPS: A Program that Simulates Human Thought, pages 279–293. McGraw-Hill, 1963.

[170] Allen Newell and Herbert A. Simon. Computer science as empirical enquiry. *Communications of the ACM*, 19:113–126, 1976.

[171] Nils J. Nilsson. Eye on the prize. *AI Magazine*, 16(2):9–17, 1995.

[172] Timothy James Forester Norman. *Motivation-Based Direction of Planning Attention in Agents with Goal Autonomy*. PhD thesis, University College London, London, 1996.

[173] Randall C. OReilly. *The Leabra Model of Neural Interactions and Learning in the Neocortex*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1996.

[174] Jeanne Ellis Ormrod. *Human Learning*. Prentice-Hall, fourth edition, 2003.

[175] Andrew Ortony and Terence J. Turner. What is basic about basic emotions? *Psychological Review*, 97(3):315–331, 1990.

[176] Judea Pearl. *Causality*. 2000, Cambridge, Cambridge University Press.

[177] Adam Pease. Standard Upper Ontology Knowledge Interchange Format, 2004.

[178] Barney Pell, Douglas E. Bernard, Steven A. Chien, Erann Gat, Nicola Muscettola, P. Pandurang Nayak, Michael D. Wagner, and Brian C. Williams. An Autonomous Spacecraft Agent Prototype. In *The First International Conference on Autonomous Agents*, pages 253–261. ACM Press, 1997.

[179] David Pereira, Eugènio Oliveira, Nelma Moreira, and Lus Sarmento. Towards an Architecture for Emotional BDI Agents. In *The Twelfth Portuguese Conference on Artificial Intelligence*, pages 40–46, 2005.

[180] Rosalind W. Picard. *Affective Computing*. MIT Press, London, 1997.

[181] Edgar Allan Poe. *Maelzel's Chess Player*. Amazon Press, 2004.

[182] Martha E. Pollack and Marc Ringuette. Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In *The Eighth National Conference on Artificial Intelligence*, Boston, Massachusetts, 1990.

[183] Ashwin Ram, Ronald C. Arkin, Kenneth Moorman, and Russell J. Clark. Case-Based Reactive Navigation:A Case-Based Method for On-Line Selection and Adaptation of Reactive Control Parameters. Technical report, Georgia Institute of Technology, 1992.

[184] Anand S. Rao. AgentSpeak(L): BDI Agents Speak out in a Logical Computable Language. In *The Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1996.

[185] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *The Knowledge Representation and Reasoning*, pages 473–484, San Mateo, California, 1991. Morgan Kaufmann Publishers.

[186] Eric Rasmusen. *An Introduction to Game Theory*. Blackwell Publishers, Malden, Massachusetts, third edition, 2001.

[187] D. Raj Reddy, Leed D. Erman, Richard D. Fennell, and Richard B. Neely. The Hearsay-I Speech Understanding System: An Example of the Recognition Process. In *The Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 185–193, California, 1973.

[188] W. Scott Neal Reilly. *Believable Social and Emotional Agents*. PhD thesis, Carnegie Melon University, Pittsburgh, Pennsylvania, 1996.

[189] Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, 2001.

[190] Steve G. Romaniuk. Evolutionary Agent Societies. In *Conference Proceeding*, 1998.

[191] Jeffrey S. Rosenschein. *Rational Interaction: Cooperation Among Intelligent Agents*. PhD thesis, Stanford University, Stanford, California, 1985.

[192] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Boston, Massachusetts, 1994.

[193] Stanley J. Rosenschein and Leslie Pack Kaelbling. The Synthesis of Digital Machines with Provable Epistemic Properties. In *The Theoretical Aspects of Reasoning About Knowledge*, pages 268–272, San Mateo, California, 1986. Morgan Kaufmann.

[194] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, second edition, 2002.

[195] Earl D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.

[196] Earl D. Sacerdoti. The Non-Linear Nature of Plans. In *The Fourth International Joint Conference on Artificial Intelligence*, pages 206–214, Tbilisi, 1975.

[197] Rui Sadio, Gonalo Tavares, Rodrigo Ventura, and Luis Custodio. An Emotion-Based Agent Architecture Application with Real Robots. In *The AAAI Fall Symposium on Emotional and Intelligent II*, pages 117–122, 2001.

[198] John Searle. *Speech Acts*. Cambridge University Press, 1969.

[199] Murray Shanahan. A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition*, 15:433–449, 2006.

[200] Claude E. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 41(314), 1950.

[201] Stuart C. Shapiro and Haythem O. Ismail. Anchoring in a grounded layered architecture with integrated reasoning. *Robotics and Autonomous Systems*, 43(2-3):97–108, 2003.

[202] N. Shardlow. Action and agency in cognitive science. Master's thesis, University of Manchester, Manchester, 1990.

[203] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

[204] Munindar P. Singh. Towards a Theory of Situated Know-How. In *The Ninth European Conference on Artificial Intelligence*, pages 604–609, Stockholm, 1990.

[205] Munindar P. Singh. Group Ability and Structure. In *The Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds*, page 127176, Amsterdam, 1991. Elsevier Science Publishers.

[206] Munindar P. Singh. Towards a Formal Theory of Communication for Multi-Agent Systems. In *The Twelfth International Joint Conference on Artificial Intelligence*, Sydney, 1991.

[207] Munindar P. Singh. A Critical Examination of the Cohen-Levesque Theory of Intention. In *The Tenth European Conference on Artificial Intelligence*, page 364178, Vienna, 1992. John Wiley and Sons.

[208] Christos Sioutis and Jeffrey Tweedale. Agent cooperation and collaboration. *Lecture Notes in Computer Science*, 4252:464–471, 2006.

[209] Aaron Sloman. *Foundations of Rational Agency*, chapter What sort of Architecture is Required for a Human-Like Agent?, pages 35–52. Springer, 1999.

[210] Aaron Sloman. Beyond shallow models of emotion. *Cognitive Processing*, 2(1):177–198, 2001.

[211] Aaron Sloman. Varieties of Affect and the CogAff Architecture Schema. In *The Symposium on Emotion, Cognition, and Affective Computing*, 2001.

[212] Aaron Sloman and Brian Logan. Architectures and Tools for Human-Like Agents. In *The Second European Conference on Cognitive Modelling*, pages 58–65, 1998.

[213] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.

[214] Stephen P. Stich. Could man be an irrational animal? *Synthese*, 64(1):115–135, 1985.

[215] Harry Stack Sullivan. *The Interpersonal Theory of Psychiatry.* Norton, New York, 1953.

[216] Ron Sun. *A Tutorial on CLARION.* Cognitive Science Department, Rensselaer Polytechnic Institute, fifth edition, 2003.

[217] Ron Sun. Motivational representations within a computational cognitive architecture. *Cognitive Computation*, 1:91–103, 2009.

[218] Ron Sun, Edward Merrill, and Todd Peterson. From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, 25:203–244, 2001.

[219] Ron Sun and Isaac Naveh. Simulating organizational decision-making using a cognitively realistic agent model. *Journal of Artificial Societies and Social Simulation*, 7(3), 2004.

[220] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.

[221] Katia P. Sycara. Argumentation: Planning Other Agents' Plans. In *The Proceedings of 11th International Joint Conference on AI*, pages 517–523, 1989.

[222] Edward Lee Thorndike. *Animal Intelligence: An Experimental Study of the Associative Processes in Animals*. The Macmillan Company, 1998.

[223] Larissa Z. Tiedens and Colin Wayne Leach. *The Social Life of Emotions*. Cambridge University Press, 2004.

[224] Alan Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[225] Manuela Veloso, Jaime Carbonell, Alicia Perez, Daniel Borrajo, Eugene Fink, and Jim Blythe. Integrating planning and learning: The PRODIGY architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 7(1):81–120, 1995.

[226] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.

[227] H. Vigneron. Les automates. *La Nature*, 2142, 1914.

[228] Victor Harold Vroom. *Work and Motivation*. Wiley, New York, 1964.

[229] Ipke Wachsmuth. *Prerational Intelligence: Adaptive Behavior and Intelligent Systems without Symbols and Logic*, volume 1, chapter The Concept of Intelligence in AI, pages 43–55. Kluwer Academic Publishers, 2000.

[230] Gerd Wagner and Florin Tulba. Agent-oriented modeling and agent-based simulation. *Conceptual Modeling for Novel Application Domains*, 2814:205–216, 2003.

[231] Yingxu Wang. A cognitive informatics reference model of autonomous agent systems (aas). *International Journal of Cognitive Informatics and Natural Intelligence*, 3(1):1–16, 200.

[232] Yingxu Wang. On the cognitive processes of human perception with emotions, motivations, and attitudes. *International Journal of Cognitive Informatics and Natural Intelligence*, 1(4):1–13, 2007.

[233] Yingxu Wang, Ying Wang, Shushma Patel, and Dilip Patel. A layered reference model of the brain (lrmb). *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 36(2):124–133, 2006.

[234] Hans Weigand and Virginia Dignum. I am autonomous, you are autonomous. *Lecture Notes in Computer Science*, 2969:227–236, 2004.

[235] Sharon Wood. *Planning and Decision Making in Dynamic Domains*. Ellis Horwood, Chichester, 1993.

[236] John Woodfill and Ramin Zabih. An Architecture for Action with Selective Attention. Technical report, Stanford Logic Group, 1990.

[237] Michael Wooldridge. *Reasoning About Rational Agents*. MIT Press, 2000.

[238] Michael Wooldridge. *An Introduction to Multi Agent Systems*. John Wiley and Sons, first edition, 2002.

[239] Michael Wooldridge and Nicholas R. Jennings. Agent theories, architectures, and languages: A survey. *Intelligent Agents*, 890:1–22, 1995.

[240] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[241] Ian Wright, Aaron Sloman, and Luc Beaudoin. Towards a design-based analysis of emotional episodes. *Philosophy, Psychiatry and Psychology*, 3(2):101–126, 1996.

[242] V. J. Wukmir. *Emocin y sufrimiento: Endoantropologa elemental*. Labor, Barcelona, 1967.