SOFTWARE AGENT'S DECISION MAKING APPROACH BASED ON GAME THEORY

[1]Anju Rathi, [2]Namita Khurana, [3]Akshatha. P. S, [4]Pooja Rani
[1, 2&3]Department of Computer Science, KIIT College of Engineering, Gurgaon, India.
[4]Department of Information Technology, ITM, Sector-23 A, Gurgaon, India.

ABSTRACT

This paper highlights the use of software agent which is capable of perceiving its environment and performs its own operations without any other explicit instruction. The main objective of this paper is to investigate the possibility how to make the decision making capability of an Expert system more accurate and successful goal-oriented. This is done with the use of a utility measurement which is explained in a given example. It helps the agent to take a decision quicker so goal will be achieved.

KEYWORDS: *Software Agents, Game Theory, Expert System, Decision Making, Utility Value.*

I. INTRODUCTION OF SOFTWARE AGENT

Software agents are autonomous pieces of software that conduct several tasks delegated to them. In the era of endless information flows, benefits can be achieved by authorizing certain kind of tasks to be done automatically by small independent software programs. Software agents are continuously running, personalized and semi-autonomous, and this makes them useful for a wide variety of information and process management tasks.

Numerous definitions exist for software agent and there is no single commonly accepted one. Some of them are: Software agent is a software entity that functions continuously and autonomously in a particular environment, which may contain another agents and processes. Intelligent agents continuously perform three functions as:

- Perception of dynamic conditions in the environment;
- Action to affect conditions in the environment; and
- Reasoning to interpret perceptions, solve problems, draw inferences, and determine actions

II. SOFTWARE AGENT AND ENVIRONMENT

On the basis of definitions, it is noticed that a software agent is a piece of software that is able to act autonomously in particular environment. Figure1 from Wooldridge [3] illustrates how agent interacts with its environment. However, there is no commonly accepted definition by many researchers. Generally, it is stated that an agent is a software entity that is able to conduct information-related tasks without human supervision [1], which can be viewed as an autonomic property of an agent which shows that software agent have decision making capability before taking any action[2] as shown below in the figure 1:

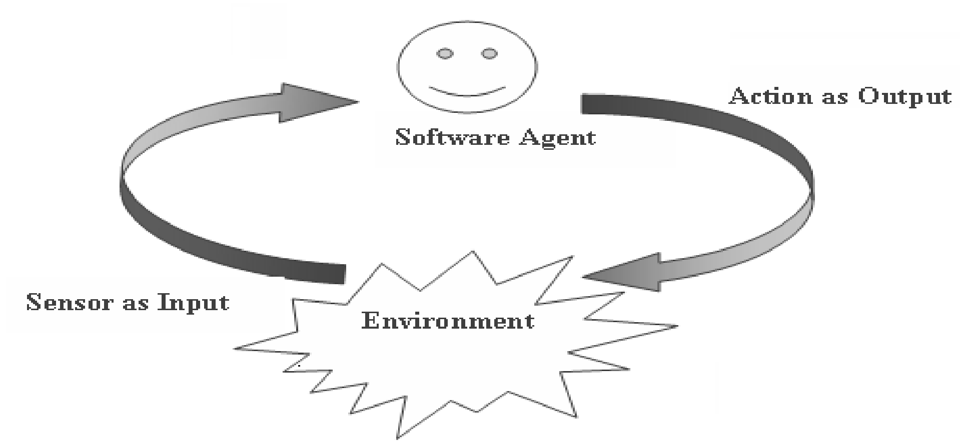


Figure 1: Software Agents and its Environment

III. MAIN FEATURES OF SOFTWARE AGENTS

- **Reactivity** Agent should be able to perceive its environment and respond to changes that occur in it. It shows its reactive behavior based on some changes in input.
- **Proactivity** An agent should also have the ability to take the initiative and not only react to external signals. This helps agent to pursue its individual goals (goal-directed behavior).
- **Cooperation** An agent should be able to interact with other agents. This can be arranged via agent-communication language.
- **Learning** shows that agents have to learn new things when they interact with external environment. It increases the performance of the agents with the collection of new knowledge gained.

IV. DIFFERENT TYPES OF SOFTWARE AGENTS

There are various types of software agents which are noticed by many researchers and all have their own role to play. Some of them are shown in figure 2 which are as:

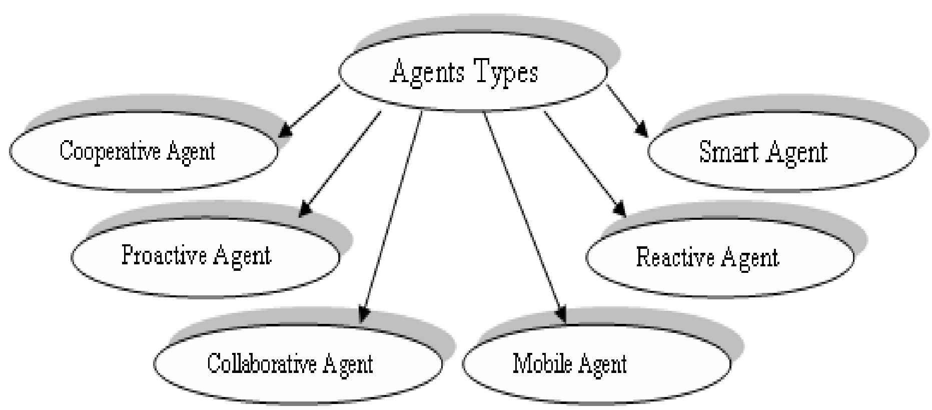


Figure 2: Types of Software Agents

4.1 Cooperative Agents communicate with other agents and give their next response based on that communication.

4.2 Proactive Agents initiate actions without user prompting.

4.3 Collaborative Agents are proactive and cooperative with other agents. They share their information with others within the same environment or group.

4.4 Mobile Agents are those who are able to migrate from host to host to work in a heterogeneous network environment.

4.5 Reactive Agents are reactive in nature, senses input and take decision for the specific tasks for which these are responsible.

4.6 Smart Agents are autonomous, cooperative and learner in nature so able to perform its tasks efficiently and smartly.

Multi-Agent System (MAS)

Multi-Agent Systems (MAS) are systems composed of multiple agents and these multiple agents work together to achieve the main objective.

V. SOFTWARE AGENT: HOW TO TAKE DECISIONS

To become a responsive and proactive agent there is requirement of decision making capability. For example, a responsive agent responds to the environment inputs it senses. But to do so, the agent must decide how to respond as well as the most appropriate time to respond, whether it does so immediately or has time to analyze the situation. Furthermore, a proactive agent is able to take action without being specifically prompted to, if it senses an opportune scenario. Clearly this capability requires an agent be able to decide both when to take action as well as what action to take.

Furthermore, beyond simply making a decision, not all decisions are good decisions. Consequently decision making protocols are often analyzed and compared by parameters such as: negotiation time, simplicity, stability, social welfare, pareto efficiency, individual rationality, computational efficiency, and distribution and communication efficiency. In terms of negotiation time, it is clearly not useful for an agent to take exceedingly long periods of time to make a decision such that the decision making mechanism cannot be used in practical situations.

An unstable design mechanism does not repeatedly arrive at the same conclusion in identical scenarios. Consequently, with unpredictable choices, an unstable design mechanism cannot be trusted to represent the user of the software agent. Social Welfare is a measure of the overall value of all agents as a sum of each agent's payoff.

Pareto efficiency also views the overall global perspective in which no alternate solution benefits any individual agent. Individual rationality on the other hand pertains to each agent individually rather than collectively. For an agent to be individually rational, the resulting payoff from a decision must be no less than the payoff an agent receives by not participating in whatever the decision at hand may be. If an agent is not computationally efficient, it cannot be implemented in a realistic setting and is effectively useless. Similarly, if communication between agents and the distribution of processing between multi-agent systems is not efficient then the system will be subject to computational limitations and may not necessarily be a useful decision making mechanism [7].

One type of decision making theory is Subjective Expected Utility (SEU), which is a mathematical technique of economics that specifies conditions for ideal utility maximization. However SEU deals only with decision making and does not describe how to model problems, order preferences, or create new alternatives. Furthermore, SEU theory requires strong assumptions such as the consequences of all alternatives are attainable, and as a result it cannot be applied to complex real problems [8].

Rational Choice is an economic theory based upon a hypothetical `economic man' who is cognizant of his environment and uses that knowledge to arrange the desired order of possible actions. However, much like SEU, rational choice theory falls short as a complete decision making model because it does not specify how to perform the calculations necessary to order choices [10].

Welfare economics analyzes the effect of resource distribution amongst members of the society as a whole. The aforementioned social welfare is a measure of welfare economics which seeks to maximize the average utility of each member in the society. A similar concept is egalitarian social welfare which seeks to maximize the value of the worst member of the society. However, there are limitations which restrict the satisfiability of the members. Fundamental desirable properties of a social choice rule are:

- Existence of a preference ordering for all possible choices which is defined for all outcome pairs,
- An asymmetric and transitive ordering,
- Pareto efficient outcome,
- Independence of irrelevant alternatives, and
- No single agent dictator dominating the preferences of others.

VI. WHAT IS GAME THEORY?

Game theory is a branch of mathematics that aims to lay out in some way the outcomes of strategic situations. It has applications in politics, inter-personal relationships, biology, philosophy, artificial intelligence, economics, and other disciplines. John von Neumann is looked at as the father of modern game theory, largely for the work he laid out in his seminal 1944 book, Theory of Games and Economic Behavior, but many other theorists, such as John Nash and John Maynard Smith, have advanced the discipline which is applied in many applications because its scope has greatly increased recently.

VII. DEFINITION OF GAME THEORY

In the form of economics, it is a theory of competition stated in terms of gains and losses among opposing players. As defined by Parsons and Woolridge game theory as "studies interactions between self-interested agents". In particular it studies the problems of how interaction strategies can be designed that will maximize the welfare of an agent in a multi-agent encounter, and how protocols or mechanisms can be designed that have certain desirable properties"[9]. Game theory is also described as a collection of techniques to analyze the interaction between decision- makers using mathematics to formally represent ideas. Thus, game theory serves as a technique which attempts to compute an optimal choice amongst several in a strategic interaction. One of the driving tenants of game theory is that in a game of a two players, opponents must seek to minimize their potential for loss while maximizing their potential for benefit [9].

VIII. LIMITATIONS OF GAME THEORY

The most limiting constraint of game theory's applicability to general multi-agent decision making is the computational efficiency evaluation criteria of decision making mechanisms. To function as a generalized decision mechanism, a game theoretic agent would have to be able to adapt to varying input requirements, opposing players, different rule sets, and unique preference relations for each game and set of players. While game theory defines various solution techniques, some of which are optimal, the solution concept varies among games and there does not exist a single solution approach applicable on all the games.

Furthermore, game theory focuses upon the existence of solution concepts but does not specify the algorithmic techniques necessary to compute the solutions. Consequently, many game theoretic techniques assume unlimited computational resources and are often NP-Hard problems [9]. And thus, neither the computational efficiency nor the negotiation time constraints are necessarily satisfied by a game theoretic decision making mechanism.

IX. MAKING SOFTWARE AGENTS AS AN EXPERT SYSTEM

9.1 What is an Expert System?

Expert Systems are computer programs which play the role of an expert related to their domain. The main goal is to understand the situation and take intelligent decisions like human beings. The fundamental principle underlying an expert system is to embed domain specific knowledge regarding how to solve a particular problem within a production system such that it may reason and attempt to devise a solution with a quantifiable confidence in that decision. Additionally, an expert system also embodies the ability to explain its reasoning by responding to `how' and `why' queries by the user. The knowledge base and explanation subsystem illustrated in **Figure 3** supply the extensive domain knowledge and provide the justification of the systems reasoning respectively. The inference engine performs the reasoning for the expert system just as the control structure of a production system selects among the applicable production rules.

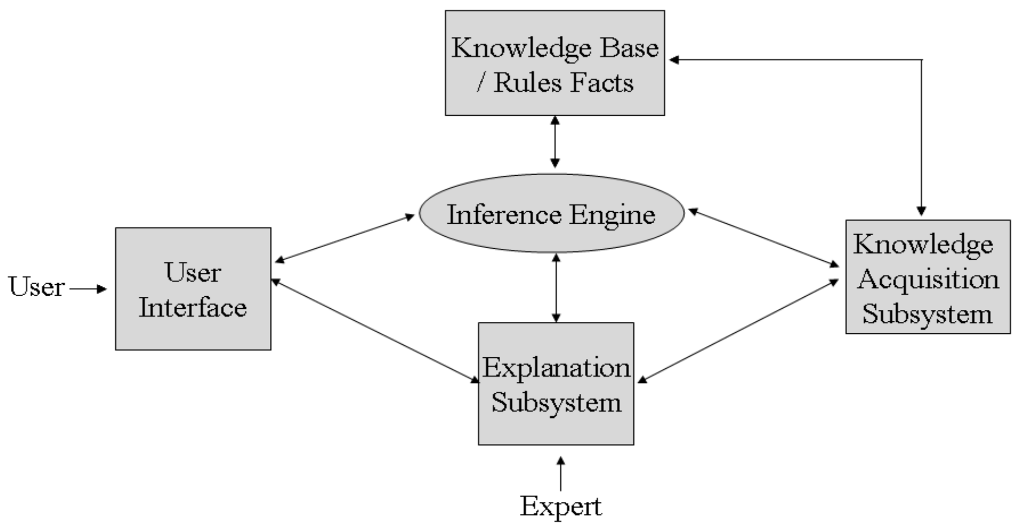


Figure 3: Expert System Architecture

9.2 Expert System as a Smart Agent in Making Decisions

It is noticed that Expert System has characteristics like software agents. Kay and McCarthy's notion of a software agent carrying out a goal and asking for advice if necessary is fundamental to the function of an expert system. As already described, an expert system is designed to try and solve a particular problem, and furthermore does so without additional interaction unless the user must be queried to obtain information necessary to make a decision.

The characteristic of being situated is satisfied by the general software agent capability of perceiving the environment in the form of askable information and acting upon the environment by the selected decision signal. The askable information may be obtained from either another agent capable of answering queries or a human user and thus an expert system is also social.

Once an expert system has been programmed with the knowledge of a subject matter expert, and obtained all of the necessary information from the environment to guide its decision making process the resulting decision is obtained autonomously without further intervention. To be flexible, an expert system would need to be responsive and proactive. Although an expert system may respond to the signals it receives and requires to make informed decisions, it cannot proactively analyze the environment and decide when to take action. Effectively, although an expert system does not meet all of the generally agreed upon guidelines which describe an intelligent software agent, the majority of the criteria are met and therefore an expert system may be regarded as a rudimentary intelligent agent.

9.3 How to Enhance the Decision Making Approach

The efficiency of the system performance is dependent upon the inference engine's search algorithm so a good searching algorithm should be applied. Main of them are as:

- **Rule-based expert system** performs a goal-driven state space search and
- **Case-based expert system** search for the most similar case to the current scenario.

When searching among possible options there are various techniques such as a depth-first or breadth-first search.

- A **breadth-first search** examines all of the possible options at the current state of the system. A breadth first search implements goal-driven state space search, and consequently it is most efficient if the first line of reasoning traversed leads to a solution so the system does not have to backtrack and try another possible sequence of actions.
- A **depth-first search** on the other hand traverses the state space graph along a single line of reasoning until a terminal node is met either corresponding to a successful conclusion, otherwise if the line of reasoning is unsuccessful it then backtracks to consider the next line of reasoning.

Alternatively, in this paper we propose game-theoretic concepts of utility measures may be incorporated to enhance the decision making performance of expert systems. Depending upon the implementation, the utility measure is not necessarily tied to a particular payoff, but rather is a preference ordering by which the possible actions are arranged in order of desired outcome. The particular utility values assigned to the knowledge set are implementation dependent.

9.4 Example: Basketball Game Expert Coach

We have created a rudimentary basketball coach system termed Coach Rule-base Expert System (CRES), which calls an offensive play according to specific game situations and player capabilities. Game of basketball serves as a beneficial example because it embodies several meaningful criteria such as diametrically opposed players, a complex decision domain, and a two level decision hierarchy by which first the defense selects a play and then the offense calls a play. Here we have four situations where a smart agent takes the decisions autonomously which are as:

1. **Last minute game situation** is defined as when there is less than a minute left in the game and the offensive team is down by two points or less. The coach would also know the capabilities of his own players, and so if he has good jump shooters he will go for the win and attempt a three point shot. Otherwise, rather than hoping for a lucky bounce, if the coach does not have exceptional three point shooters he will instead call a play looking for a two point basket tying the game so that they may go for the win in overtime.
2. **A man-to-man defense** is defined as one in which a single defender is guarding each player regardless of where they move on the court, and that there are not multiple defenders guarding any single player. Depending upon the skills of the offensive players on the floor there are two possible plays:
   - If there are good jump shooters on the offense, then the coach calls for the offense to set a screen opening up a jump shot for one of the players. Due to the fact that each defender is focusing on an assigned offensive player, they are most likely guarding the good offensive player tightly, so by setting a screen for a teammate, an offensive player may be freed for a high percentage jump shot.
   - If the offensive team does not have good jump shooters playing at the particular point in the game but the defense is guarding them in a man-to-man defense, the coach calls for a pick and roll in which the offensive players work together to move closer to the basket for a better percentage shot.
3. **A zone defense** approach in which a player guards a particular region of the floor rather than a particular player. Here, the coach recognizes a zone defense by the fact that the same defender is not following offensive players across the floor and there are not multiple defenders guarding a single offensive player. Once again depending upon the individual capabilities of the offensive players there are two basic plays the coach may call:
   - If the offensive team has good outside shooters, then it is recommended to attack the zone defense by shooting long distance jumpers that the zone is not adequately guarding against. The spacing of the defenders in a zone defense to cover the floor opens up the inside for a dominant low post player to post up.

- If the offense does not have exception jump shooters, they should pass the ball in for a post player to get a closer higher percentage shot.

4. **Double Team** is applied when the offensive team has a star player the defense may double team the individual player. A double team is easily recognized by the coach, in which there is not a single defender, but rather multiple players simultaneously guarding the star player. Here, another player on the offensive team is not being guarded, and the coach advises the team to pass the ball to the open player who is not being guarded and effectively the highest percentage shot.
   This expert coach system is designed to vary its performance in accordance to the abilities of the players it is emulating such as whether the team consists of skilled jump shooters or post players. This example is based upon the perspective that the offensive team has excellent jump shooters. The utility measure decision making enhancement is incorporated by strategically ordering the rules based upon average utility.

The Utility Preference shows utility ordering assigned for each possible defense based on their chances to win the game with their possible situations and their possible winning directions which are shown in the Table 1 below:

Table 1: Utility Preference

Offense/ Defense	Last Minute	Man Def.	Zone Def.	Double Team	Sum	Average	Utility
Win_on_3	7	4	4	5	20	5	5
Tie overtime	6	3	2	3	14	3.5	3
Screen shot	5	6	5	1	17	4.25	4
Free_player	4	7	7	7	25	6.25	7
Shoot Jumper	3	5	6	6	20	5	6
Pick Roll	2	2	3	2	9	2.25	2
Post Up	1	1	1	4	7	1.75	1

It is noticed that if utility value is used then it saves time and move in the right goal-based direction where the possibility of achieving the goal is maximum. So as shown in figure 4 goal is more closer and team is in good state as actions taken by the Coach as an expert system based on decisions which are made on the basis of utility value. The utility enhanced ruled ordering approach arrived at a solution in only its second line of reasoning.

When decisions are not based on their utility value then decisions would be taken randomly as shown in figure 5 where goal is achieved after taking many steps not goal-directed conformation is here. This random order decision making approach arrived at a solution in its fifth line of reasoning.

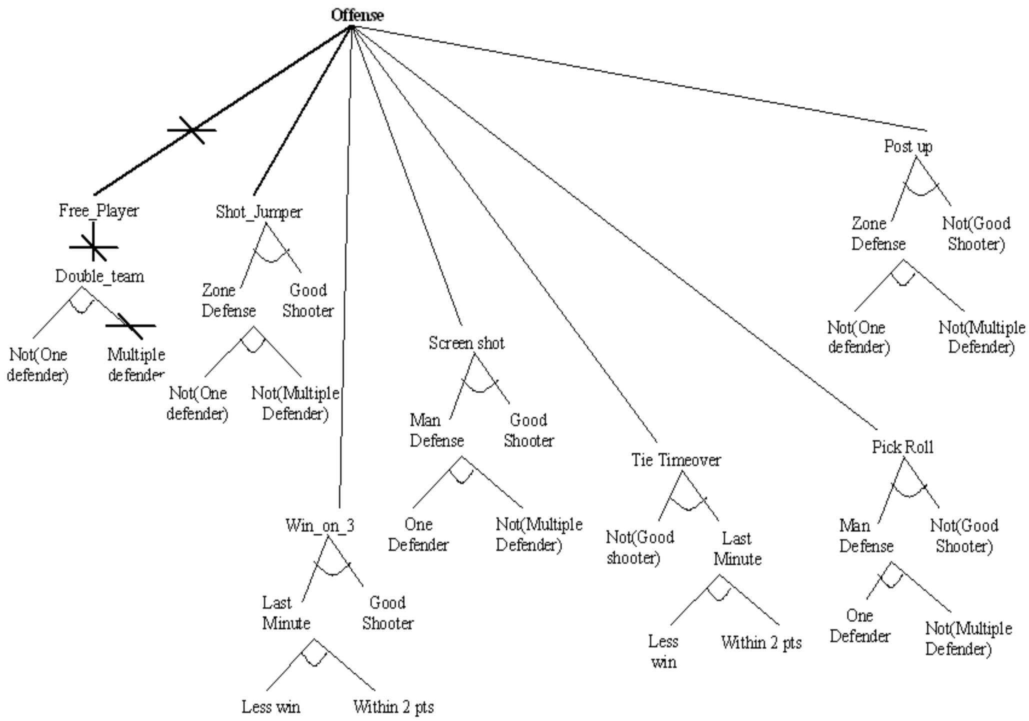


Figure 4: Utility Enhanced Decision Making

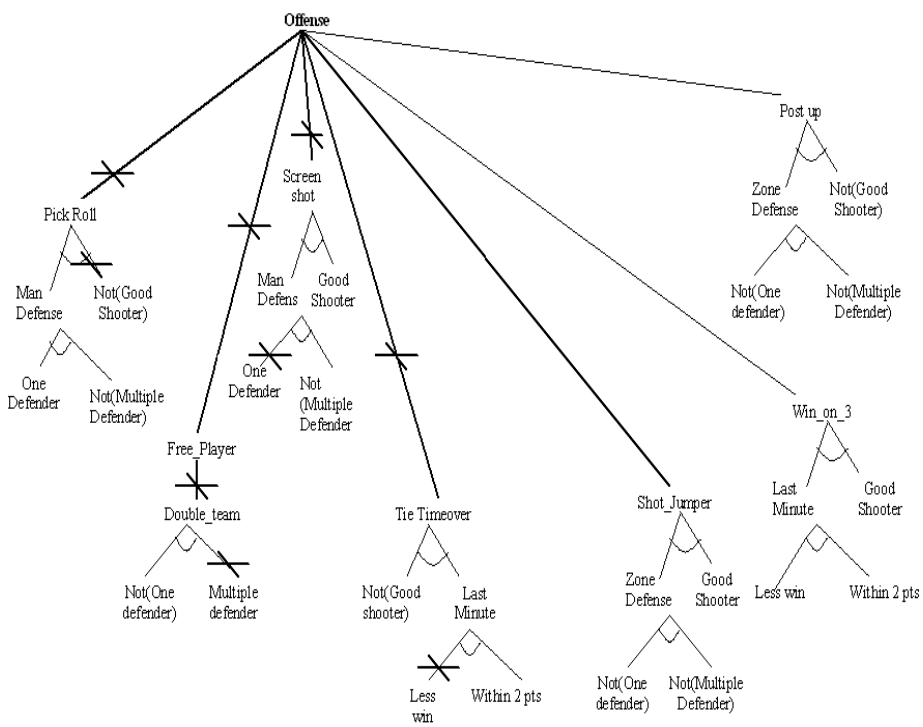


Figure 5: Random Ordered Decision Making

Note: The bold lines mark the lines of reasoning attempted moving left to right, with X's denoting eliminated options.

X. CONCLUSION

Making an intelligent and good decision is not an easy task. A smart agent achieves the given goal with minimum instructions from others and has the capability to reason and make decisions related to that domain. Here game theoretic utility notions are used to enhance the decision making mechanism of an expert system. Doing so provides a more strategic means of improving expert system performance rather than relying upon inconsistent heuristic approaches. Thus, making decisions is no easy task but doing so in a more strategic rational manner has greater value than making random choices.

XI. FUTURE WORK

Here a relatively simple rule based system approach is used so it can be expanded with a larger rule based system to incorporate more specific games and a defensive expert system could be coupled to oppose the offensive expert system in which case analysis can be performed on the possible result by the opposing system. A video game with multiple graphical effects and an expert system with good artificial intelligence could be applied for good game application.

REFERENCES

[1] M. Boman, S.J. Johansson, and Lyback D. Parrondo strategies for artificial traders. In Proceedings of the 2nd International Conference on Intelligent Agent Technology. World Scientific, October 2001

[2] Brooks, R. A. (1991c), "Intelligence without Reason", *In Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Menlo Park, CA: Morgan Kaufmann.

[3] E. Durfee. Coordination of Distributed Problem Solvers. Kluwer Academic Press, Boston, 1988.

[4] Elaine Rich and Kevin Knight. Artificial Intelligence. McGraw Hill, second edition, 1991.

[5] Evan Hurwitz and Tshilidzi Marwala. Multi-agent modeling using intelligent agents in the game of lerpa, 2007.

[6] George F Luger. Artificial Intelligence Structures and Strategies for Complex Problem Solving. Addison Wesley, fifth edition, 2005

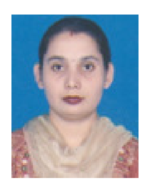
[7] Herbert A. Simon. A behavioral model of rational choice. The Quarterly Journal of Economics, 1955.

[8] Herbert A. Simon and Associates. Decision making and problem solving. http://www.dieoff.org, 1986

[9] Martin J. Osborne and Ariel Rubinstein. A Course in Game Theory, 1994.

[10] Ulle Endriss. Multiagent systems: Rational decision making and negotiation. http://www.doc.ic.ac.uk/ue/mas 2005

Authors Biography



Anju Rathi was born at Faridabad, Haryana, India in 1981. She has done her graduation in 2002 from the Maharishi Dayanand University, M.C.A in 2005 from M. D. University, Rohtak & M. Tech from M. D. University, Rohtak. Her Research interests include Genetic Algorithm, Artificial Intelligence and Software Engineering.

Namita Khurana was born at Hansi, Haryana, India in 1981. She has done her graduation in 2001 from the Kurukshetra University, M.C.A in 2004 from G.J.U University Hisar, M.Phil in 2007-08 from C.D.L.U, Sirsa & pursuing M.Tech from Karnataka State University. Her research interests include Soft computing , Artificial Intelligence.

Akshatha.P.S was born at Kolar district, Karnataka, India in 1983. She has done her B.Tech in SJCIT, Karnataka. She is now pursuing M.Tech in Lingaya's university, Faridabad. Her research interests are Computer Networks, Database Management System.

Pooja Rani was born at Moujpur (Delhi) in 1982. She has done her B.C.A. in 2004 from IGNOU, Delhi, MCA in 2008 from IGNOU, Delhi & M.Tech (SE) from ITM, Gurgaon (MDU). Her Research interests include Software Engineering, Java, RDBMS, and Networking.