Development of Artificial Intelligent Skills and Techniques in Agricultural Robotics

G. Govinda Rajulu1*, Vijayakumar Maragal Venkatamuni2 and P. Manimegalai1

1Karpagam University, Coimbatore – 641021, India; govinda_sai@yahoo.co.in, manimegalai.vairavan@gmail.com
2RPRC Department of CSE, Dr Ambedkar Institute of Technology, Near Jnana Bharathi Campus, Bengaluru 560056, Karnataka, India; dr.vijay.research@gmail.com

Abstract

Objectives: This analysis paper mainly focuses on the development of cognitive architecture where the agents at different levels exhibit different levels of thinking and on the acquisition of smart sensor motor skills which is known as Construction Skill Tree (CST). Methods/Statistical Analysis: These concepts are implemented through simulation by using prolog programming language. This simulation is an imitation of the operation of real world fruit picking robot system over time. It includes the discovery of one’s own body, including its structure and dynamics. This includes the acquisition of associated cognitive skills such as self and non-self-distinction. Finding: The obtained simulation results can be given by designing and implementing the Construction Skill tree Implementation Architecture (CSIA). The design of CSIA provides faster skill acquisition. Hence it is called CSIA, the CSIA has a five layer and first four layers are single agent environment. Improvement/Application: The proposed cognitive architecture has collections of agents that work together for reaching predefined goal. In CSIA these contains reflexive, reactive, deliberative, thinking and meta-thinking layer.

Keywords: Agent, Construction Skill tree Implementation Architecture (CSIA), Cognitive Architecture, Smart Fruit Picking Robots

1. Introduction

Agriculture and Industry are the backbone of the Indian economy and more than 60 % people work in the agricultural land day and night irrespective of season and climate. India is also a developing country where we can find a large population working below the poverty line. So as to meet their ends every day they involve in some or the other hazardous labor work. This project was aimed at developing an intelligent Robot that could be capable of doing all the work of a labor under any seasonal and climatically worst conditional over the world construction workers are exposed to a number of hazards at work and due to work. The hazards can be:

1. Physical hazards and injuries because of noise and vibration, extreme cold and heat, working in windy, rainy or foggy Conditions, exposure to ultra violet radiation and electric welding.
2. Chemical hazards such as dusts, fume, mists and gases.
3. Ergonomic issues and degenerative issues.
4. Biological hazards.
5. Psycho-social hazards.

So there is a need to overcome the above mentioned few of the issues and the Labor Robots can be an alternative and of course a solution for the above said issues. It is not that we are changing the entire labor sector and providing an irreversible Alternative. It is just an alternative which can be applied whenever

*Author for correspondence
and wherever required. In this context, this project was taken up to overcome hazardous labour work. Robotics is defined as a creation of intelligent mechanical devices which can cope with the complexities of the real world. In\textsuperscript{1,2} Today’s robots are systems which have high degree of complexity. The complexity depends on their function, components used, computer hardware and software designed\textsuperscript{3,4}. Test beds and benchmarks are mainly using in the field of robotics for comparing architectures and outcomes. Robotics is one of the important areas of cognitive sciences. For example EM-ONE\textsuperscript{5,6} architecture, in the cognitive architectures is also classic example for robotics. This research does not differentiate biological (human and non-human minds) or machine mind.

2. CSIA Architecture

Experimental Setup

The proposed architecture has 5 layers as shown in the Figure 1. It is the extension of Swarm Ambient Cognitive Architecture (SACA) architecture. The CSIA architecture has 5 layers, and first four layers are single agent environment and in each layer the agent goes on learning or feeding its skills set which reflects on the knowledge of the agent. Everywhere a certain action is carried out by an agent. All these actions of agent will be stored as subtasks. When the agent has to perform a larger task, it will use these subtasks to achieve the main task. In this way a skill tree will be constructed. This increases the performance rate of an agent.

Language used for implementation is java. Java was selected for programming as it is compatible to run in different platforms, robustness, high performance and object oriented. Net beans is used for The NetBeans Platform is a framework for simplifying the development of Java programs. NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components. Applications based on the NetBeans Platform, including the Net Beans integrated development environment. Applications can install modules dynamically.

3. Modules

The different modules as shown in the Figure 2 are: Agent Creation; Parameters; Agent Movement; Agent Energy Consumption; Agent Communication.

![Figure 2. Schematic diagram.](image)

3.1 Agent Creation

By using the java applet codes the agents are created. Agents are the intelligent actors who perform the actions based on the task defined to the agent. Agents are in oval shape and the movements are defined. As per the proposed architecture there are five layers, different agents are defined for different layers and all the agent navigates through a pre-defined path.

3.2 Parameters

The parameters are “Obstacle”, "rotten fruit", “unripe fruit” and “ripe fruit”. These are differentiated based on the color. Obstacles are represented using cyan color and the rotten fruit is given by red color, unripe fruit is given by green fruit and yellow color represents the ripened fruit. And these are all represented in the
square shape and the same color formats are followed throughout the research implementation.

3.3 Agent Movement

The agent in the environment has to do the particular task, like collecting the fruit. To do so, the agent is given with the movement. The path is restricted to the straight line in the simulation and first four layers are given a single agent within the environment. The agent moves in the Testbed and reflects on the output it does and the action result is considered accordingly. But in the last layer the travelling path is given to all over the environment. Whenever each movement or action behavior is executed, it is replicated on the agent and carried out for the further layers. Here the “avoid” behavior of the agent in reflexive layer is carried out in all the higher layers.

3.4 Agent Energy Consumption

Agent navigates using “wander” and “avoid” behaviors. If an agent encounters a “fruit”, it executes “react” behavior. “React” behavior of the robot is shown in the form of “energy”. In simulation this increase or decrease in energy is shown in the form of agent’s size. If the agent’s energy is increased, it is shown by increasing the size of the agent. If the agent’s energy is decreased, it is shown by decreasing the size of the agent.

3.5 Agent Communication

The main implementation is mainly concentrated on agent communication with other agents in the environment. From each layer, the agent keeps building on action and action result. In first four layers agent learns its behavior whenever it encounters a fruit or an obstacle. This behavior is executed whenever the same situation is encountered again. Main implementation is given by communication between agents in the environment. We are showing it as the dynamic representation and the idea of implementation is the area where the agents implied will be brought to the network communication, as this brings up the overcoming of abnormal scenario where it will not be able to distinguish between ripe and unripe fruits due to weather changes. The solution we tried to give is completely based on color, energy parameter and shape parameter. Whenever the energy criteria is not replicated on size parameter, it will be reflected on the network, where the result will be depicted on all the agents as the action, action result and location will be stored, so that the next agent when it arrives to the field, it will never move towards that food as it is confirmed by one agent that it is not the one to collect, and thus the efficiency is improved.

3.6 At the Reflexive Layer

The word reflex is an action that is performed without conscious thought as a response to a stimulus, and considering it the agent will be left on the Testbed. The agent is given with the straight path movements (as per our consideration) and it is asked to move towards the destination. The agent moves in the same path and as the reflex behavior says, it’s like a part of response to the stimulus in the environment. The agent is asked to move towards the destination in this. And the reflexes is built as in such, whenever it finds an obstacle it show the reflex nature of taking a diversion or moving away precisely to avoid an obstacle. Whatever may be the obstacle it should move away from the obstacle and reach its destination? As it is a single agent environment, each time the agent comes this reflexes will be the first step to store. And in this many considerations can be taken where the agent number is restricted to the count of 1, and it has to reach its destination. Here the obstacle is given by the cyan color, and whenever it finds it, it should move away from it. Agent should also be able to encounter the obstacle count, if finds more than one obstacle it should avoid all the obstacles and reach the destination, if no obstacle is found it should directly reach the destination. Our code implementation shows one agent moving towards the destination where the obstacle count is 2 and the agent path is restricted to straight line. Even if the destination’s position is changed, the agent will follow the input it has got and accordingly navigate towards destination. Finally the agent will learn to avoid the obstacle, reach to all destinations and in prior the change in any of the positions is implicated on the entire agent and the knowledge is built. This can be implemented in the real world scenario, whenever the robot moves in the field and the destination is a particular tree, the robot has to reach the tree. If it encounters any obstacle it will avoid and move towards the destination. This will be the Initial stage of learning and this knowledge will be reflected in all the layers.
3.7 At the Reactive Layer

The word “reaction” means response to stimulus. In\(^2\) Here the agent's response with respect to a particular fruit is depicted. This is also a single agent Environment and the agent will show different response or reaction when it encounters the fruit. Our main consideration is given as color priority, whenever it encounters those color, the response will be stored. Whenever the agent encounters the particular cultured fruit, the action result is depicted. Here to show it has got some result, we have used the energy resemblance but this is just a parameter consideration, and this in real terms is considered as the performance issue. When the agent enters the test bed, it should reflect the knowledge learnt from the previous layer and it will avoid the obstacles wherever it is found and moves towards the destination given as trees. When an agent encounters fruits, it will check the Color of the fruit. There are three kinds of fruits provided our scenario. It should collect all the three types of fruits as it will be a reaction to the stimulus. When it finds the red color (rotten fruit), the reaction for this is given as the decrease in size of the agent. When the agent finds a fruit with green color (unripe fruit), the reaction for this is given as the increase in size of the agent by “n” units. When agent finds the fruit in yellow color (ripe fruit), the reaction for this is given as the increase in size of the agent by “2n” units. All these reactions of an agent will be stored by the agent. In this layer, the agent learns the type of the fruit and the consequence of actions if it collects the fruit. In\(^3\),\(^4\) the code implementation a single color is given to a fruit and the consequence of actions if it collects the fruit. By “2n” units. All these reactions of an agent will be stored. Whenever the agent encounters the fruit, it will check the Color of the fruit. There are three kinds of fruits provided our scenario. It should collect all the three types of fruits as it will be a reaction to the stimulus. When it finds the red color (rotten fruit), the reaction for this is given as the decrease in size of the agent. When the agent finds a fruit with green color (unripe fruit), the reaction for this is given as the increase in size of the agent by “n” units. When agent finds the fruit in yellow color (ripe fruit), the reaction for this is given as the increase in size of the agent by “2n” units. All these reactions of an agent will be stored by the agent. In this layer, the agent learns the type of the fruit and the consequence of actions if it collects the fruit. In\(^3\),\(^4\) the code implementation a single color is given to a particular type of fruit. So the agent will learn to identify the fruit based on its color. It knows the consequence if it collects a particular type of fruit. This depicts internal state of an agent which is the knowledge gained by the agent in this layer. This addition of intelligent behavior is carried forward in the next layers.

3.8 At the Deliberative Layer

At the deliberative layer means relating to or intended for consideration or discussion. Here in\(^5\) this the internal state means the knowledge it learnt from the previous layers and the energy status or the performance or the internal state of the agent is depicted here and the agent's avoid behavior is carried further. The agent with lesser capacity to hold fruits can move till a threshold value, here we consider it as first threshold. The agent moves until the first threshold and finds for the fruits. If rotten fruit is found, it is avoided as the agent knows that collecting rotten fruit had decreased its energy. This is learnt by the agent in reactive layer. The action learnt in\(^6\) reactive layer is carried forward to the deliberative layer. The agent is able to decide on its own based on its internal state. The internal state of an agent refers to the knowledge the agent has learnt in the previous layers. The agent with more capacity to hold fruits can move towards the second threshold. It searches for fruits there. It avoids rotten fruit if found. It collects the other two types of fruits. Thus the agent is capable of decision making. Based on its internal state, that is the knowledge gained in the lower layers, the agent has learnt to avoid collecting the fruit that had decreased its energy metric. Hence it avoids collecting rotten or red cultured fruit. And the agent always collects unripe or ripened fruit. In our code the idea is given such the three fruits are considered and again a single agent at a time and the thresholds are mentioned by the line bars and the two thresholds are considered for the small and big agent respectively. Considering the situation when the small agent (agent with lesser capacity) comes as this performance is very less it should stop at the provided threshold, if any of the fruits are found, the small agent collects the nearest fruit either yellow or green avoiding red. If no fruits are found then it should stop at the threshold. When the big agent (agent with high capacity) comes the action is, it should move towards nearest fruits avoiding the red color fruit and if no fruits are available in the near distance the big agent has to move towards the far distance that is the second threshold and collect for the green or yellow fruit and out of that which would be very near. In our code though changing the colors of the fruit, the agent is able to recognize the red color fruit and able to avoid it anytime. Many considerations are taken such, different three color fruits, at different thresholds, at different count of fruits at different size of agent, first two rotten fruit, all rotten fruits, changing the priority of the fruit colors. As all of the scenarios are taken and considerations are made according to our knowledge, we are able to execute 12 successful test cases.

3.9 At the Thinking Layer

At the thinking layer - The meaning of this is, the process of consideration or reasoning\(^7\) about and this is the layer where the agent communicates to itself realizing the internal state on performance rate. As it started avoiding the rotten fruit and obstacles, thinking factor is added for
agent. Agent thinks on its own realizing that more reward is found only for the ripened fruit and this thinking is reflected by avoiding the unripe fruit. Describing this, it avoids the green fruit as energy rate was only $n$ units realizing the need of yellow fruit. This thinking structure of agent overcomes the scenario of real time by avoiding the rotten fruit and the unripe fruit where the attention will be given to only ripe fruit and asked for the yield or the efficiency will be more and it overcomes the disadvantage of picking all kind of fruits. In\textsuperscript{15} the implementation, the code runs as such, this is also taken as the single agent environment provided the three different color fruits, depicting the knowledge from the old layers, the avoidance of red colored fruit will remain the same and it aims for achieving further efficiency. The agent thinks on its own to achieve $2n$ energy though it finds a green color fruit in less distance. The whole idea is such that unripe fruit should be avoided and the task of the agent is only to go to the yellow fruit, where the yield would increase. In our code the action is such though three colors of fruits are available and however red will be avoided as deliberative result and here green fruit is avoided as it has made itself to move towards the yellow color. Though the green fruit is found near instead of yellow, it will travel till yellow. Provided the real time scenario, the present fruit picking robots runs with man handling but implementing this, the action could be performed remotely and agent will just pick the ripe fruit instead of rotten or unripe.

### 3.10 At the Meta Thinking Layer

This layer describes thinking on thinking. In\textsuperscript{16} thinking layer, the agent learnt that it has to collect only ripened fruits. But due to change in weather or climate conditions, a ripened fruit might look like an unripe fruit. And the agent might not pick that fruit thinking it is an improper fruit. This leads to a decrease in performance by agent. Hence Survey on the fruit picking robot and fruit say that the fruits under abnormal conditions will not completely resemble the ripe fruit as in case they will differ a little in shape or color or density. And taking this as the basic task we started feeding the agents. Consider the abnormal condition as the yellow colored fruit with different shape. When an agent comes towards it and collects it, according to the reactive layer there should be changes on energy, but as the shape of the fruit is not matching their arises the ambiguity “whether to pick that fruit or not to pick”. The agent is not sure of its decision. Such cases will be stored and that particular fruit is kept as last priority. Communication here is not a direct communication with the agent to agent; it is done through global array where this happens in the networked area. To make this happen we implemented the idea of building a network area in the field where agents will be deployed and the all agents will be monitored at one hand. The advantage of this is that there is no need of man power as this monitoring comes under the advance monitoring. Here we raise one more advantage of reducing the human intervention. As the agent is in attach with network influence, the agent when it found itself in a different situation which is not cultivated for it, it would trigger a global array message which will relay on the network and this will reflect on all agents. Under such exceptional conditions, the agent tries to put a message via global array with these parameters - “location, action and action result”, the network passes this message to all the agents present in the environment, and hence the next agent which wants to navigate will not move towards that fruit and the efficiency is achieved. Thus handling abnormal conditions and monitoring the agents is met.

### 3.11 CST in Meta Thinking Layer

CST is a hierarchical reinforcement learning algorithm which can build skill trees from a set of sample solutions trajectories obtained from demonstration. CST uses an incremental Maximum A Posterior (MAP) change point detection algorithm to segment each demonstration trajectory into skills and integrate the results into a skill tree. CST was introduced by George Konidaris, Scott Kuindersma, Andrew Barto and Roderic Grupen in constructing skill trees is the idea implemented in the Meta thinking layer, the algorithm is implemented in this to learn each and every step that is achieved in each layer. CST forms the hierarchical learning and the first four layers where the agent is implemented on single environment considerations, so in\textsuperscript{17,18} each layer it has to learn the results of the action it performs. Here the main focus is kept on the “Avoid behaviour”, where in the first layer i.e reflexive layer, the agent learns to avoid the obstacle and moves towards the destination, and the same will be carried to reactive layer where it avoids obstacle and moves towards the destination, and the reaction carried on each fruit is registered here. In\textsuperscript{19,20} the deliberative layer, the agent learns to avoid rotten fruit as the agent knows the reaction when it collects the rotten fruit. In\textsuperscript{21} thinking
the agent avoids the unripe fruit and thus the performance of agent is improved. And in meta thinking all the skills are carried forward where the hierarchy of actions are taken forward and when an ab

4. Experimental Results

As shown in Figure 3 Robotic simulation the planned fruit picking agent collects 21 goals in 400 cycles by renewing the energy compare to unplanned agent which collects only 3 goals and lives for only 111 life cycles. The experiment was conducted for 500 life cycles to find out the in-depth potential of the micro agents through their lifespan. The planned fruit picking agents collects 21 goals in 400 cycles by renewing the energy compare to unplanned agent which collects only 3 goals and lives for only 111 life cycles. This result proves that goal based micro agent can reason about their change of aims (deliberation) watch their status (self-regulation or self-control) and achieve their goals.

![Performance Comparison](image)

**Figure 3.** Performance comparisons.

5. Society Benefits

The research aims in implementing construction skill tree in meta-thinking layer. So that we can use the skills acquired in some other problem to more quickly solve new problem. This will add a higher level intelligence to the agent and hence helps in distinguishing between ripe fruit, unripe fruit and rotten fruit. It also helps in overcoming few other disadvantages. This work mainly concentrates on developing higher level intelligence to the agent, so that it provides better efficiency and will be effective in distinguishing between fruits that are ripe and unripe; this can be achieved by implementing meta-thinking layer and adding CST in it which helps in faster skill acquisition. Farmers cannot work in hazard environment but the robots can work in hazards environment. Also higher overall availability of robot workers (no lunch breaks or vacations) and many more, to complete large amount of work in less time. There by it helps the farmers.

6. Conclusion

This cognitive architecture provides the agents the capacity of making decisions and the ability to learn based on the consequences of previous actions. The agents are able to distinguish between the obstacles and different kinds of fruits based on the color. This concept can be implemented in fruit picking robots, which will help the robot to distinguish between ripe and unripe fruits. And if the agents come across abnormal conditions, the condition is stored which consists of the position of the fruit, shape of the fruit. This is communicated to other agents. This helps the further agents to not to move towards such fruit. This results in an increase in the performance of other agents. This can also be implemented in smart fruit-picking robots, which enhances the efficiency of existing smart fruit-picking robots.

7. References

1. Davis DN. Vision of Mind, Architecture for Cognitive Diversity, Idea Group (IGI), 2005, p. 340.
2. Miri H. CernoCAMAL: A Probabilistic Computational Cognitive Architecture. Lambert Academic Publishing, 2012, p. 200.
3. Wellman HW, Cross D, Watson. Metaanalysis of Theory-of-Mind Development, The Truth about False Belief. National Center for Biotechnology Information, U.S. National Library of Medicine. 2001 May-Jun; 72(3):655−84.
4. Georgeff M, Pell B, Pollack M, Tambe M, Wooldridge M. The Belief Desire-Intention Model of Agency Proceedings of Agents, Theories, Architectures and Languages, Springer Berlin Heidelberg, 1999, 1555, p. 1−10.
5. Venkatamuni VM. A Society of Mind Approach to Cognition and Metacognition in Cognitive Architecture, PhD, Computer Science, University of Hull, Aug 2008.
6. Konidaris GD. Autonomous Skill Acquisition. PhD, Computer Science, University of Massachusetts, May 2011.
7. Konidaris GD. CST-Constructing Skill Trees by Demonstration, Doctoral Dissertations, University of Massachusetts, Ahmrest, 2011.
8. Miri H. CernoCAMAL: A Probabilistic Computational Cognitive Architecture, The University of Hull, UK, 2012.
9. Wellman HM, Cross D, Watson J. Metaanalysis of Theory-of-Mind Development: The Truth about False Belief, Child Development. 2001; 72(3):655–84.
10. Gwatkin JJ. Robo-CAMAL: Anchoring in a Cognitive Robot, PhD Thesis, Department of Computer Science, University of Hull, 2009.
11. Venkatamuni VM. A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture, PhD Thesis, Computer Science, University of Hull, 2008.
12. Sing P. EM-ONE: Architecture for Reflexive Common Sense Thinking, PhD Thesis, Massachusetts Institute of Technology, Cambridge: Massachusetts, United States, 2005.
13. Singh P, Minsky M. An Architecture for Combining Ways to Think. Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems, Cambridge: MA, 2003.
14. Minsky M, Singh P, Sloman A. The St. Thomas Common Sense Symposium: Designing Architectures for Human-Level Intelligence, 2004; 25(2):113–24.
15. Michael G, Barney P, Martha P, Milind T, Michael W. The Belief Desire-Intention Model of Agents, Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures and Languages Springer-verlag, 1999, p. 1–10.
16. Toan P, Michael W, Lin P. Learning within the BDI Framework: An Empirical Analysis. Book title: in Knowledge-Based Intelligent Information and Engineering Systems, 2005; 3683:282–88.
17. Alejandro GH, Fallah-Seghrouchni AE, Soldano H. Learning in BDI Multi-agent Systems, Computational Logic in Multi-Agent Systems, 2004.
18. Rao AS, Georgeff MP. Formal Models And Decision Procedures for Multi-Agent Systems, Technical Note, AAI1, 1995.
19. Georgeff M, Pell B, Pollack ME, Tambe M, Michael, Wooldridge. The Belief-Desire-Intention Model of Agency, Intelligent Agents V: Agents Theories, Architectures, and Languages, Springer-Verlag, 1999.
20. Alexander P, Braubach L. Winfried Lamersdorf. Jadex: A BDI Reasoning Engine.Multi-Agent Programming, Birkhäuser Basel, 2005, p. 143–68.
21. Wooldridge MJ. Reasoning about Rational Agents, MIT Press, 2000.
22. Rajendran M, Kannan TR, Sathish V, Durgadevi M. The Relevance of Communication Skills to the Technical Students in the Colleges of Engineering and Technology: A Study, Indian Journal of Science and Technology. 2016 May; 9(19):1–3.