iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** I **Month of publication:** January 2020

DOI: http://doi.org/10.22214/ijraset.2020.1081

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

Enhancing the Adaptation of a Rational Agent

Ibuomo R. Tebepah[1], Friday E. Onuodu[2]
[1]Computer and Mathematical Science, University of Africa, Toru-Orua, Bayelsa State, Nigeria
[2] Computer Science, University of Port Harcourt, Nigeria

Abstract: Rational Agents are computing systems (artificial intelligent systems) that display certain level of knowledge based on their programming. The knowledge is gathered from the sensors (actuators), informing the agent of its environment and how it is expected to respond or react. What if we want the rational agent to perform an undefined action; it implies that it may most likely give a default respond. The research presents the use of stored procedure as a means of enhancing the functionality of a rational agent. Data from the actuators will automatically trigger the procedure every time an undefined action is initiated, and action performed when similar has is taken again.

Keywords Rational Agent, Artificial Intelligence, Adaptation, Robots, stored procedure

I. INTRODUCTION

Agent is anything which can perceive its environment through sensors and acting upon it through actuators based on its perception [1]. In other words, an agent run in cycles of perceiving, thinking and taking actions. The fact that it has the ability of perceive, think and take actions makes such an agent a rational agent. Thus rational agent work with previously acquired knowledge, and its performance is usually measured based on prior knowledge. Rational agents are created to be autonomous [2] and directs its activities towards achieving the set goals. This is the reason behind using Belief-Desire-Intention (BDI) [3] which is a practical and popular cognitive framework for implementing practical reasoning in computer programs that has led to the development of agent programming. BDI model makes a computing system or object a rational agent within the context in which its behaviour is defined and how it should respond. However, this limits capacity of the rational agent behaviour since it uses a rule-based agent [4] programming language even though it may be performing optimally based on its programming.

With increase in the expectations, demands and usage of rational agent for more sophisticated task in different industries, rational agents are now expected to live beyond the rule-based. It is expected that once deployed, it should have the capacity to adapt and respond to "unprogrammed" task by some inbuilt mechanisms that will cause it to take appropriate actions as though it was originally programmed to perform the task (percept).

The next few sections of the work covers a review of related works, identified approaches for improving adaptation of an agent, simulated evidence, discussions and conclusion.

II. AIM AND OBJECTIVES

This paper is aimed at enhancing the adaptation of a rational agent.

The objectives of this research are to;

A. Design a rational agent that performs a specified task
B. Implement a stored procedure in the agent that enables it to perform

III. RELATED WORKS

Rationality of an agent at any given time depends on the performance measure, agents prior knowledge, expected actions to be taking or performed, and agents percept sequence to date [5]

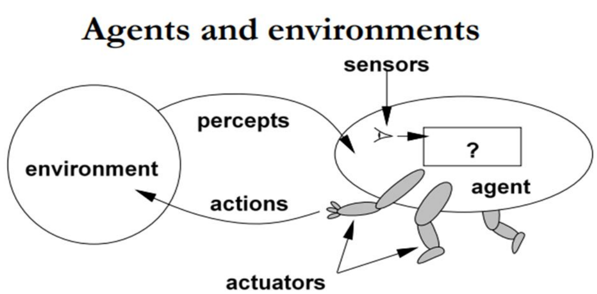


Fig 1: simple agent framework; Source: Stuart Russell and Peter Norvig 1995

In the above we see simple design of a single agent environment. The percepts is sent to the sensors, stimulating a corresponding action via the actuators back to the environment. These actions though automated in a way is also not automated in another because for each possible percept sequence, a rational agent is to select an expected action that maximise its performance measure given the evidence provided by the percept sequence and whatever built in knowledge the agent possess. So if we assume or create a means through which knowledge can continuously be added to an agent externally or internally, sending to the sensors that activate the actuators, such an agent will naturally adapt to changes and we should have achieve high level of autonomy. In any case, an intelligent agent should not only gather information but also learn as the agent gains experience. This may be inform of modified augmented – a priori (self-discovering through reasoning).

Rusell's [6] research identified different rationalities – the perfect rationality which constraints an agent's action to provide maximum expectation of success given the information available. It is concerned with knowledge level analysis in AI system that establishes an upper bound on the performance any possible system. In order to address the challenges of the perfect rationality, the calculative rationality based on equation analysis prior to agent action was introduced. The main issue with this was the level of calculation required and trade-off. Despite these, the calculative rationality has been the mainstay of both the logical and decision theoretic traditions. This is because it guarantees high performance measure.

The next is the metalevel rationality – trade off type II rationality. The working of this type of rationality is finding the optimal trade-off between computational cost and decision quality. Type II rationality is the maximisation of expected utility taking into account deliberation cost.

The aim of this was to take advantage of the architecture and gain resources for computation, which it was able to achieve in a way but since it is a second decision-making process whose application domain comprises of object-level computation themselves and the computation object states that affect, trade-offs are always an issue.

In the final analysis, Russell proposed a bounded optimality which also entails calculative rationality. The work proposes adaptation through the content, reinforcement learning, compilation method, fixed and variable computation cost, fully variable architecture, and offline and online mechanism.

Wang [7] accurately stated that people will trust an autonomous system such as robot if they have a more accurate understanding of its decision making process [8]. The fact that human and intelligent or rational agent coexist as human's have chosen to automate daily routines, which brings or introduces trust as a critical element to how human and rational agent perform together [9]. Since each element is expected to perform within its conferment, the disuse and misuse [10] of knowledge is avoided even though this is not usually the case in real life.

Wang work simulated a game program laying emphasis on human-robot interaction as means of adaptation. The research demonstrated trust between the rational agent and the human as each one can only perform its required task with the help of the other. In essence, the agent and the human need to communicate by means of sending and receiving signals before task could be completed.

Even though this was achieve as explained in the work, the level of communication was not a multi-level interaction. For instance, an agent sends or notifies a human of impending danger, the human makes the decision to either accept or reject the warning without any further referral to the agent.

Communication is then re-established based on the actions of the human. For instance if the human failed to yield to the agent warning, the human is expected to explain afterwards to strengthen the trust between the two.

Every rational agent is expected to perform at its optimal theoretically on its knowledge level within the defined environment, and this knowledge level relies on the assumption that the agent has a perfect understanding of its environment – more of a perfect rationality.

IV. APPROACHES

To improve the adaptive capacity of an agent, its rationality level has to increase using any or a combination of these approaches

A. The Design Philosophy

This forms the architectural framework which splits into 2 notions. Firstly, the object level that carries out the conceptual concerned with the application domain, and secondly the decision making process consisting of object level computations. The correct design for a rational agent depends on the task environment – the physical environment and the performance measures on environment histories. This will determine how the agent is expected function.

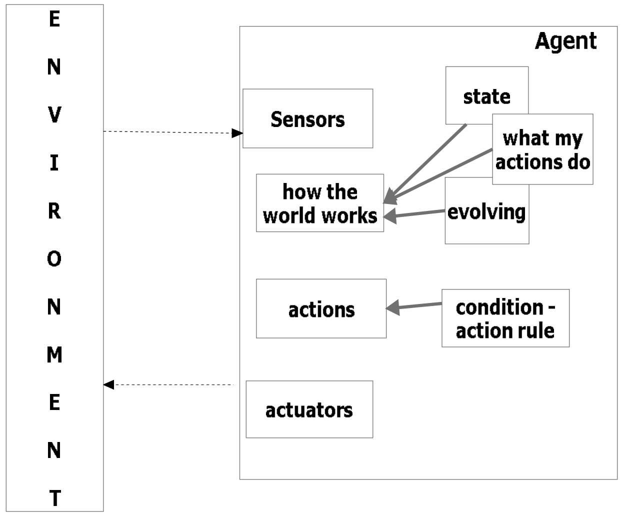


Fig 2: existing system architecture source: Stuart Russell and Peter Norvig 1995

The challenge with the design approach is that modifications will be difficult after implementation.

B. Stored Procedures

Stored procedures are sub routines or modules program that can be executed on its own or as part of an application. Stored procedures are used in computing environment to optimised systems and application. This is the recommended approach for enhancing the functionalities of rational agent.

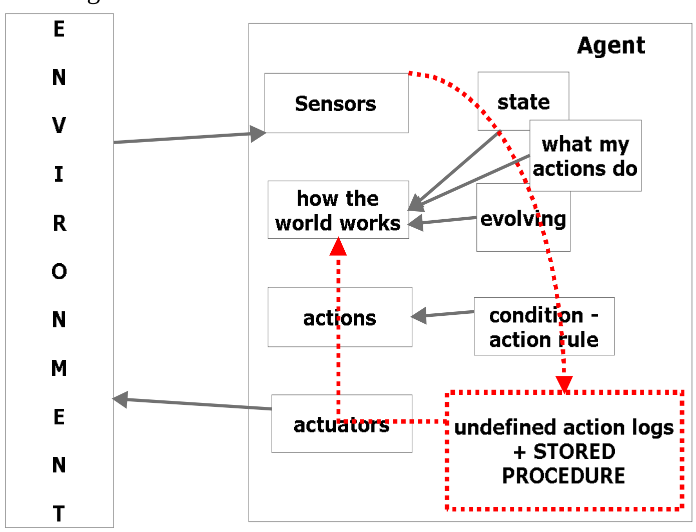


Fig 3: proposed system

V. PROCEDURE

In other to experiment the stored procedure approach, a simulation environment was created using RoboDK, text editor and Python programming language. The procedure involves series of installation such as updating graphic card driver, installation of RoboDK – a robotics simulation software for creating robotic models for the research and experiment purposes.

A. Defining Agent Goals Are

1) Primary goal in which the rational agent to make a simple painting on an object.

2) Secondary goal the agent send incorrect input data to the log database, and invoke stored procedure with correct action when next such a simpler action is taking.

RoboDK platform was used for the modelling and simulation of the rational agent (robot). It was used to define certain parameters such as environment of operation, axes, coverage or reach, and functions.

Database with a single table called sensor Data with 3 field names namely; sensor Number, sensor Name, and sensor Description to store logs of incorrect data input. This logs defines or serves as data input for the execution of the stored procedure. Data input is based on the response from the sensors which are the controllers.

a) Sensor Number: Derived from the default number attached to each sensor in the model

b) Sensor Name: Programmable name assigned to a sensor

c) Sensor Description: Role assigned to the sensor such as Left Move, Right Move etc

This is created in MySQL, and connection is made using procedure written in Python. The rational is to get data from the sensors when an undefined instruction is passed to the robot.

i) Undefined Senor Data is a stored procedure attached to the robot model that will receive data from the sensors and stores data in the sensor Data table.

ii) Invoke Data is the procedure that will be called whenever previously undefined action are performed again. The procedure uses the field names to verify, defines action, and sends it to the sensors to perform.

VI. ALGORITHM

Problem summary: enhance a rational agent actions with unlabelled data

Objective: To create and integrate a stored procedure in a rational agent

A. Rules

1) Enable the rational agent to function autonomously

2) Utilised data sent via the agent controllers (sensor)

3) Rational agent re-coordinates its actions when the procedure is invoke by a user requesting for previously incorrect action

B. Base Case 1

1) Name: undefined Sensor Data

2) Input: rational agent is asked to perform undefined action

3) Output: sends request to sensor Data

4) Description: when a rational agent user, makes a request of the rational agent to perform an undefined task like requesting to paint an undefined target, it invokes the undefined Sensor Data procedure, and details of the action such as the sensor or controller in which the request was made is sent to the sensor Data

5) Logical Process:

a) Begin

b) Moves cursor to rational agent

c) Clicks on robot tool for painting

d) Clicks on object platform to paint

e) If (click == defined target)

```
{
        Object platform painted;
}
Else
{
        undefinedSensorData is invoked;
}
End;
```

C. Base Case 2

1) Name: invokeData

2) Input: the user of the rational agent performs previously incorrect action

3) Output: invokes the invokeData procedure

4) Description: this retrieves data from the sensorData, attached an action, and sends to the sensor or controller details it retrieved

5) Logical Process

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

a) Begin
b) Moves cursor to rational agent
c) Clicks on robot tool for painting
d) Clicks on object platform to paint
e) If (click == defined target)

```
{
        Object platform painted;
}
Else
{
        invokeData is invoked;
        object platform painted;
}
End;
```

VII. RESULT AND DISCUSSION

Based on the defined objectives of the rational agent, the simulation showed movement and execution of action expected which is the painting of a platform. Every anticipated movement is coordinated by the 3 axis attached to robot, tool and objects to be painted. The platform 1 is the initial target object for painting. Platform 2, though within the environment of the robot, it was not however predefined as the case with platform 1. Thus an attempt will trigger the undefined Sensor Data procedure to be executed.

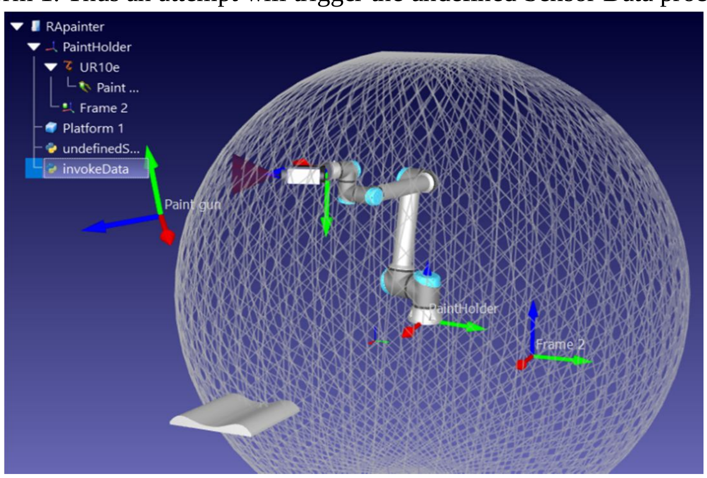


Fig 4: robot and defined object (platform 1)

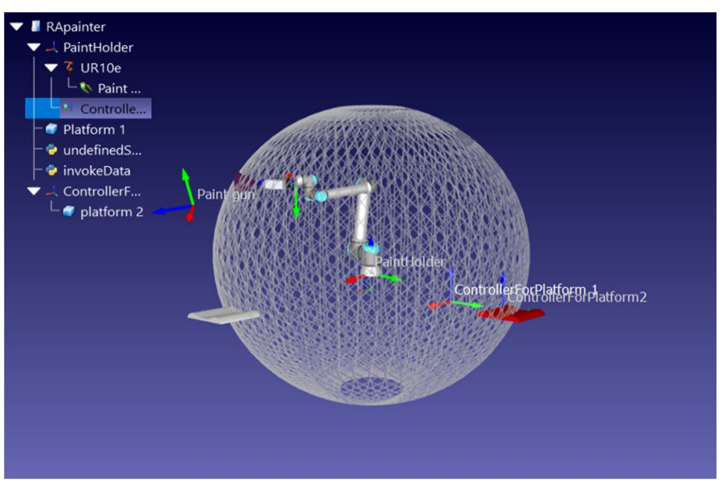


Fig 5: introduction of untargeted object (platform 2)

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

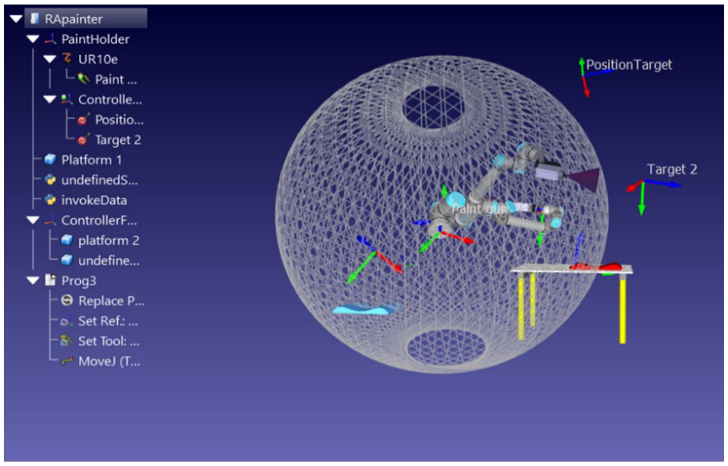


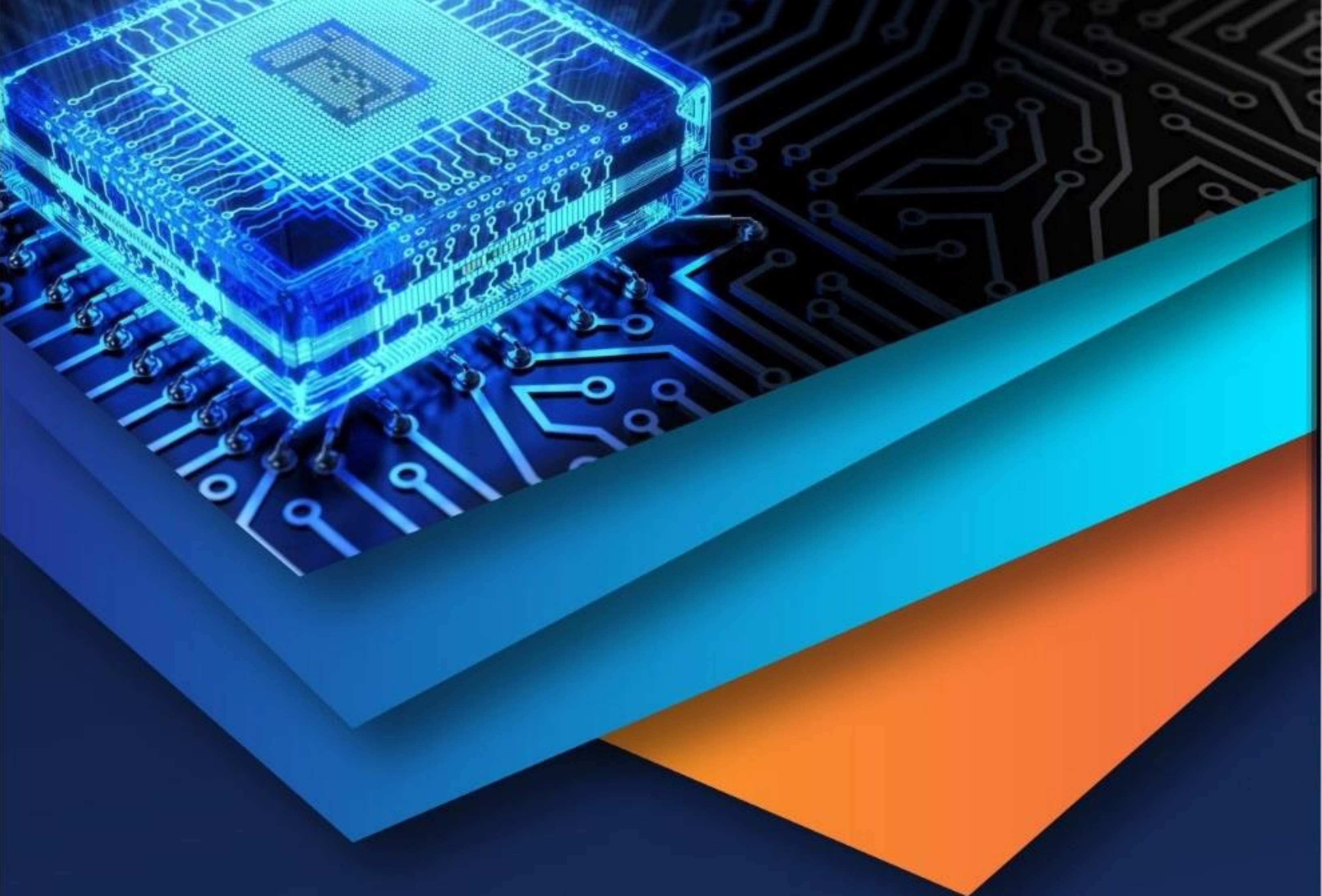
Fig 6: working on platform 2

VIII. CONCLUSION

In conclusion, the simulation showed different aspects of robot functionalities and the limitless possibilities. However, certain limitations such as the using a trial version of RoboDK hindered very precise coordination, saving of projects. Also, it clearly showed that except for a controlled environment or for research purposes, the use of stored procedures, though enhances the adaptability of rational agent, it is also prone to abuse and breach of security which may lead to malfunctioning of the rational agent. This then opens more avenues for further research.

REFERENCES

[1] Stuart Russell and Peter Norvig (1995), Artificial Intelligence: A Modern Approach, Prentice-Hall
[2] Elmahalawy, A.M (2012), Intelligent Agent and Multi-Agent System, Journal of Engineering and Technology
[3] Imam, I., and Kodratiff,Y., (1997) Intelligent Adaptive Agents, AAAI AI Magazine 18 (3)
[4] Lacey, N., and Heximor, H., (2012), A Framework for Coherence Based Multi-Agent Adaptivity, ResearchGate
[5] Imberman, S (2010), Using Robotics to Create an Intelligent Agent available at: http://www.cs.csi.cuny.edu/~imberman/ccli/robotagents.pdf
[6] Russell, S., (1997), Rationality and Intelligence, Artificial Intelligence, 94
[7] Wang, N., Pynadath, D., Unikrishanan K.V., Shankar, S., and Mechant, C., (2015), Intelligent Agents for Virtual Simulation of Human-Robot Interaction, Springer
[8] Lee, J., and Moray N.,(1992), Trust Control Strategies and Allocation of Function in Human Machine System, Ergonomics 35 (10)
[9] Lee, J,and See, K.A (2004), Trust in automation: Designing for Appropriate reliance, Ergonomics 46 (1)
[10] Parasuraman, R., and Riley, V (1997), Humans and Automation: Use, Misuse, Disuse, Abuse, Ergonomics 39 (2)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY