Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.395 – 400

RESEARCH ARTICLE

DESIGN PATTERN BASED ANALYSIS IN MULTI-AGENT FRAMEWORK

A.Sindhuja[1], Rathi.P.R[2], Anis Fathima.A.R[3], A.P.Jeyasanthini[4]

[1] Asst. Professor, Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry, India
[2,3,4] UG Student, Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry, India
[1] sindhu.lect@gmail.com; [2] rathi2592@gmail.com; [3] aniszarin@gmail.com; [4] jeyasanthini@gmail.com

Abstract— **A modern machine or software can deal with some inbuilt programs which represent a solution to a problem statically where the inputs and the outputs given are constant. During a machine gets inputs at run – time the program must be able to work depending on new situations at that time it needs some intelligence or knowledge which are developed and implemented by agents. When a single agent running in an environment it may identify only some of the states but when multiple agents are running each individual state can be judged easily. Hence in our proposed system a multi agent environment is created and each set of actions are done by individual agent concurrently for a health based system. Thus the actions identified by each agent are communicated through communication languages and justify the solution for that given problem using Evolutionary algorithm and it is stored as a pattern for further use. When there is a solution for a problem in the knowledge base it can be identified through pattern based ontology which is applied on multi agent environment. Thus it may decrease the time to find new solutions.**

Keywords— **Multi agent system; e-health care; medical information system; Agents; Design Pattern based Ontology.**

I. INTRODUCTION

Artificial Intelligence (AI) is generally defined as the science of making computers do things or action that require intelligence when it is done by humans. A.I is the study of ideas that enable computers to be intelligent. A field computer science and engineering concerned with the computational understanding of what is commonly called intelligent behaviour, and with the creation of artifacts that exhibit such behaviour. An agent is an object that intellects its environment and acts upon it based on the current situation [4].

Agents acts through programs in an environment through its sensors to sense the environment and actuators to act in that environment. It acts constantly and separately in an environment in which other processes take place and other agents occur”. The events of agents are made using two important components:

- Sensors - sense the environment that is taken as input by the agent.
- Actuators – it make changes to the environment after some processing based on the algorithm used.

For a complex system, single agent is efficient in managing all the process activity. So, we go for Multi Agent System (MAS) where multiple agents are involved processing an activity, each agent will be performing unique task. As a result, overall performance and accuracy is optimized. Multi Agent System focuses on the specification of the Foundation for Intelligent Physical Agent (FIPA) which is responsible for interaction between the agents. The communication between the agents is done with the help of FIPA-ACL, Agent Communication Language. Agent Communication Language is the basis for message transfer between the several agents. Knowledge Query and Communication Language (KQCL) is the another standard communication language for interaction between the agents. But, FIPA-ACL is standardized by w3c consortium.

Java Agent Development Framework (JADE) is an enabling technology act as middleware for the development and run time execution of applications which are based on agent's paradigm and which can flawlessly work and interoperate both in wireless and wired environment. JADE is a platform where, the system is developed in Java programming language.

II. RELATED WORK

Few researchers in the current literature used Multi-agent system using Evolutionary Algorithm in various health systems. Also, there are several web services implemented using Ontology for efficient search and data retrieval. Web services using design pattern based ontology wherein the efficiency of the system is optimized. Conversely, none of the researchers tried to implement design pattern based ontology in a Multi Agent System by enhancing the time and accuracy of the medical data retrieval rather than enhancing the service. We sort out the work into two divisions: Multi Agent System with evolutionary algorithm and Health Care Management System with Design Pattern based Ontology.

For illustrating the development of a multi- agent system using agent-oriented software engineering (AOSE)

Methodologies. The implemented system includes different intelligent agents representing a medical organization with various functions and communication methodologies, as well as the communication with human users (patients and medical staff). The development procedure following an AOSE approach describes a prescribed way the actors, the organization, the internal behaviour, and the conversations between all partners, and as an outcome the worth of the final system is enhanced [3].

Multi-Agent systems (MAS) are suitable in many medical domains, due to the features of the problems in this area. MAS are the origin of an evolving technology that assurances to make it much at ease to design, to improve and to sustain. The system incorporates retrieving distributed health care system services in multi-agent environment to accomplish improved Quality of service in java platform. This builds an outline to schedule the convention between the patients and the suitable doctors for routine and substitute services [6].

Multi-agent systems are used in many medical domains, due to the properties of the problems in this domain. In this paper it is given that MAS are an appropriate tool to tackle health care issues, and we show some specific examples of the use of this technology in different concrete problems in this field. A number of open researches in the field are also described [5].

JADE (Java Agent Development Framework) is a middle-ware that outfits an effective agent platform and supports the advance of multi agent systems. JADE agent platform preserves great performance of a distributed agent system used with the Java programming language. In particular, its architecture for communication provides a flexible and efficient messaging. Obviously choosing the best available technology implanted within Java runtime environment. JADE uses Java implementation which consent noble runtime efficiency, software reuse, agent agility and the recognition of different agent architectures [2].

A new research model is emerging based on the Multi-Agent System (MAS) architectural framework which allows users and software agents to interact and thus cooperate within common application areas. Such a vision entails bridging the different "'views of the world" of knowledgeable agents through the commitment to common definitions of the conceptual entities and also the technical terms employed in knowledge base[12].

Multi agent systems consist of multiple autonomous entities having various information and diverging interests. This inclusive outline to the field deals a computer science viewpoint, but also lures on ideas from theory of games, semantics, economics logic, procedure exploration and values. Highlighting foundations, the authors offer a wide-ranging and rigorous handling of their matter, with thorough demonstrations of distributed problem solving, multi - agent communication, non-cooperative game theory, and learning, social choice, mechanism design, auctions, coalitional game theory, and belief, logical theories of knowledge, and other aspects of rational agency[9].

A multi-agent is a special case of the container-component architecture [11]. In this case the component entities are agents and the container is the agent atmosphere (environment) that offers agent discovery and agent communication. An agent is a dynamic goal-oriented entity that has one or more than one roles in the environment. Each role eternally executes a control loop such as:

1. Observe state of the environment
2. If environment state = goal condition, then quit

3. Select an action
4. Perform action
5. Goto 1

An agent may observe all or a part of the environment in step 1. It keeps track of what other agents are doing in the environment. Every agent may have unique goals. The required action selected in step 3 depends on the inspection done in step1. It may also depend on the actual state of the agent. This state could be simple (one-bit memory) or complex (knowledge base). Performing the action in step 4 alters the state of the environment. It may also change the agent's actual state and goals. The action may include directing messages to other agents.

JADE (Java Agent Development Framework) is a software frame which is totally implemented in Java language [8]. It shortens the implementation of multi-agent systems via an interface that meet up the terms with the FIPA specification and by a set of graphical tools that supports the correcting and implementing phases. The agent proposal can be distributed across multiple machines (which do not have the same Operating System) and the configuration can be guarded by a remote GUI. The configuration can be even modified at execution time by moving agents from one system to another whenever required. JADE is implemented in Java language and the minimum system requirement is the version 1.4 of JAVA (either the run time environment or the JDK).

The life cycle of a JADE agent follows the cycle suggested by FIPA[7]. These agents follow different states defined as follows:

1. Initiated: The agent is created but not yet registered.
2. Active: The agent is registered with a unique name. In this state created agent can communicate with other agents.
3. Suspended: In this state the active agents stop their execution for a specified interval of time as its thread is suspended.
4. Waiting: In this state the agents will be waiting for an event.
5. Deleted: After completing the event action, agent’s thread stops is execution and is deleted.
6. Transit: The agent will be moving to a new location.

The behaviour describes the actions under a given event. This behaviour of the agent is defined by using the method addBehaviour(). The diverse behaviours that the agent will execute are defined by the abstract class Behaviour[7]. The class Behaviour consists of abstract methods:

- action(): executed when the action takes place.
- done(): executed when the specified functionality requirement is achieved.

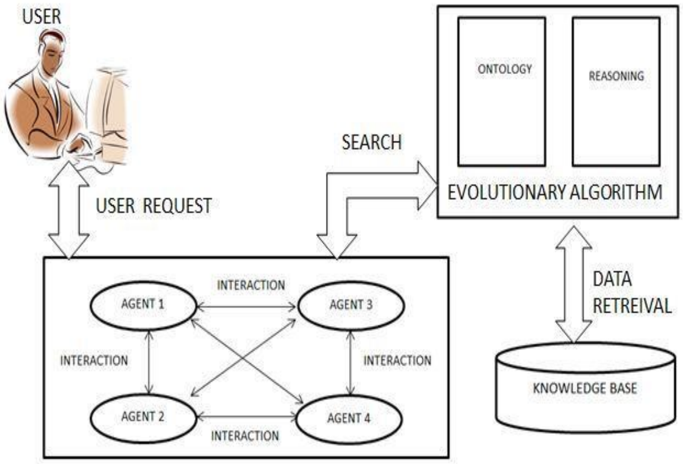
A user can override the methods onStart() and OnEnd () . In Addition, there are another two methods such as block () and restart () which is used for modifying the agent's behaviour. When an agent is locked, it can be unlocked in various ways. On the other hand, the end user can override the methods onStart() and onEnd() that the agent contains.

III. PROBLEM DEFINTION

To propose a multi – agent environment where each agent analyses some set of inputs and the related actions are found which are kept in the knowledge base of each agent. The agents in the environment will have their own knowledge base and also a search mechanism to evaluate the actions. When there is a problem arise in that environment then each individual agent communicates with its possible actions to other agents actions and decides the final solution since all agents are acting in the same environment. Hence the new solution is recorded as a pattern for future developments. If a problem has a solution which is previously stored as pattern then it can be retrieved through design pattern based ontology. Hence these features are related with a health care system where a patient detail is analysed by the initial agent, then next agent analyse the symptoms and their causes (disease name) and another agent analyse its prescription needed for further treatment. Then each individual’s actions are identified and are correlated and communicated with other agents actions if the problem is new otherwise it will depend upon previous patterns which retrieved through ontology. Thus each agent acts separately and when there is a need it will communicate with each other.

IV. SYSTEM ARCHITECTURE

The architecture of the proposed system is illustrated below. The proposed system consists of four different agents. Each agent will perform their specific tasks which are described below. The agents communicate with each other through Agent Communication language which a part of Foundation for Intelligent Physical Agent (FIPA).



Where,

Agent1 – Gets input from the user

Agent 2 – Keeps track of user given symptoms Agent 3 – Provides the prescription based on the user given symptoms from the knowledge base if present already else a new entry will be made.

Agent 4 – Retrieval and storage on the ontology.

V. RESEARCH PROPOSAL

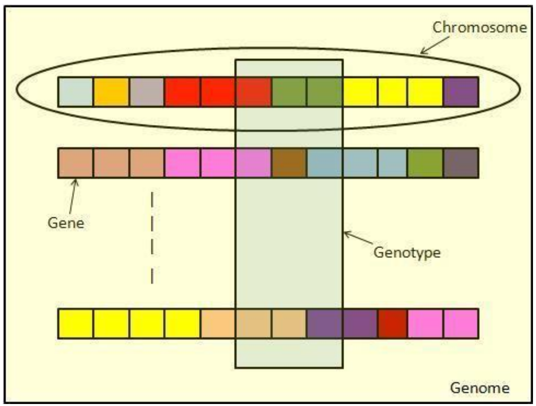
In this session, we categorized the work into two groups: Multi Agent System with evolutionary algorithm and Health Care Management System with Design Pattern based Ontology.

A. Genetic Algorithm

A genetic algorithm (or GA) is a searching technique used in computing the true or fairly accurate solutions to optimization and search problems. Genetic algorithms are classified as global search heuristics. Genetic algorithms are a meticulous class of evolutionary algorithms that use techniques inspired by evolutionary aspects such as inheritance, mutation, selection, and crossover (also known as recombination). Genetic algorithms are implemented as a simulation in which a population of abstract depictions (also known as chromosomes or the genotype or the genome) of nominee elucidations (also known as individuals, phenotypes, or creatures) to an optimization problem evolves toward better solutions. By tradition, elucidations are represented in binary format as strings of zero's and one's, but other encoding methods are also possible. The evolution usually begins with a population of randomly created individuals and happens in generations. In each generation process, the fitness of each and every individual in the population is tested, multiple individuals are selected from the current population (based on their fitness), and adjusted (recombined and possibly mutated) to form a new population. The newer population is then used in the next iteration of the algorithm. Generally, the algorithm terminates when either a extreme number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a extreme number of generations, a satisfactory solution may or may not have been reached.

- ✓ Individual - Any possible solution
- ✓ Population - Group of all individuals

- ✓ Search Space - All possible solutions to the problem
- ✓ Chromosome - Blueprint for an individual
- ✓ Trait - Possible aspect (features) of an individual
- ✓ Allele - Possible settings of trait (black, blond, etc.)
- ✓ Locus - The position of a gene on the chromosome.
- ✓ Genome - Collection of all chromosomes for an individual



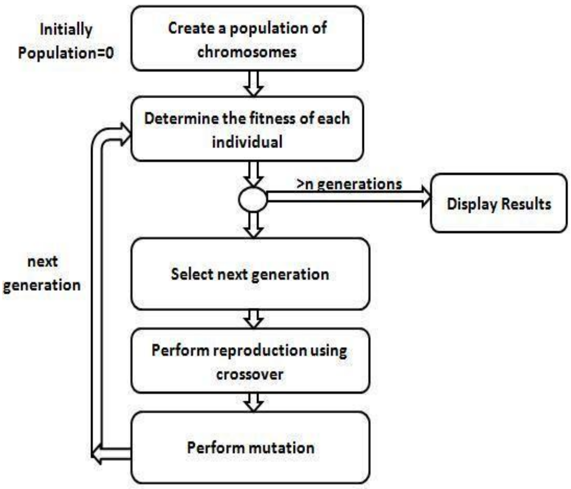
Chromosome, Genes and

Genomes Genotype and Phenotype Genotype:

– Particular set of genes in a genome Phenotype:

– Physical characteristic of the genotype (smart, beautiful, healthy, etc.)

General Algorithm for GA:


Initially
Population=0
Create a population of chromosomes
Determine the fitness of each individual
>n generations
Display Results
Select next generation
Perform reproduction using crossover
Perform mutation
next
generation

1. [Start] Generate arbitrary population of N chromosomes (Suitable way out for the problem)
2. [Fitness] Estimate the fitness *f*(x) of each chromosomes x in the pollution
3. [New Population] Create new population by repeating following steps till fresh population is complete

   [selection] Two parent chromosomes are selected from a population as per their fitness (better the fitness, the better chance is to be selected)

   [crossover] With the probability of crossover, the parents are cross over to generate a new offspring (child chromosome). If no crossover is performed, then the generated new offspring will be the exact copy of the parents.

   [Mutation] With mutation probability, mutate the new offspring at each locus (chromosome position)

   [Accepting] Place new offspring into a newest population for further use.
4. [Replace] Use newly generated population for a further execution of the algorithm

5. [Test] If the termination condition is satisfied, end and return the finest way out in current population
6. [Loop] Go to step

B. Design Pattern Based Ontology

In generic ontology, the retrieval of data takes place in static manager based on the key word provided for searching. If the data related to the specified key word is present then the data is retrieved. But, the retrieved data is only depended on the keyword and it may not be the accurate or the required data. Where as in design Pattern based Ontology, the data is retrieved based on the logic that searches the data in different aspects based on composing patterns like Logical, Reasoning, Architectural, Naming, and Re-engineering content. . Hence it reproduces an efficient search mechanism. Thus the environment can be judged on different angles to produce some meaningful facts. Conceptual Ontologies are artifacts, have a structure either linguistic or logical. Their function is to “encode” a description of the world (actual, possible, counterfactual, impossible, desired, etc.) for some purpose. Ontologies must match both domain and task, allow the description of the entities (“domain”) whose attributes and relations are concerned because of some purpose.

VI. CONCLUSION

The resulted work denotes different agents run in same environment where each agent communicates with other with their degree of knowledge with the sets of inputs. When the communication occurs with each other agents there may be a different approach of searching a procedure which is done by ontology. Hence there may be a better action which depends upon different outputs Thus it can generate more number of possible combinations of patterns which are stored in the knowledge base for further use. Therefore the system developed with multiple agents in a single environment. Hence it can be implemented in different environment with more number of possibilities of patterns.

REFERENCES

[1] queksiewkhoon.tripod.com/ontology_01

[2] F Bellifemine, G Caire and D Greenwood, “Developing Multi-Agent Systems with JADE”, Wiley Series in Agent Technology. John Wiley and Sons, 2007.

[3] David Isern and Cristina Gomez-Alonso and Antonio Moreno, Methodological Development of a Multi-Agent System in the Healthcare Domain, 2008.

[4] Artificial Intelligence [Online]. available:www.webopedia.com/TERM/A/artificial_intelligence.html, 2013

[5] Antonio Moreno, “Medical Applications of Multi-Agent Systems” Multi-Agent Systems Group (GruSMA) University Rovira i Virgili (URV) Tarragona, Spain.

[6] Srividya Bhat, Nandini S. Sidnal, Ravi S. Malashetty , Sunilkumar. S. Manvi, “Intelligent Scheduling in Health Care Domain”,IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, Sep 2011.

[7] JADE:Java Agent Development Framework [Online] available:
en.wikipedia.org/wiki/Java_Agent_Development_Framewor k.

[8] JADE:Java Agent Development Framework [Online] available: www.jade.tilab.com.

[9] Yoav Shoham, Kevin Leyton-Brown, Multi-agent System available: www.masfoundations.org, 2009.

[10] Multi-agent System, [Online] available: en.wikipedia.org/wiki/Multi-agent system.

[11] Jaron Collis, Divine Ndumu, “The Rule Modeling Guide”, [Online] available:
www.cs.sjsu.edu/pearce/modules/patterns/distArch/multi.html, 1999.

[12] Sabina Falasconi, Giordano Lanzola and Mario Stefanelli , “Using Explicit Ontologies in Agent-based Healthcare Information Systems”,1997.

[13]Ching-Seh Wu, Ibrahim Khoury, and Hemant Shah, “Optimizing Medical Data Quality Based on Multiagent Web Service Framework”, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 16, NO. 4, JULY 2012.