Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.S3, 2019, pp.54-58

DOI: https://doi.org/10.51983/ajcst-2019.8.S3.2087

Novel Classification Scheme for Multi Agents

Mohammed Ali Shaik[1], P. Praveen[2] and R. Vijaya Prakash[3]
[1]Research Scholar & Assistant Professor, [2]Assistant Professor, [3]Professor
[1,2&3]Department of Computer Science and Engineering, S R Engineering College, Warangal, Telangana, India
E-Mail: niharali@gmail.com, prawin1731@gmail.com, vijprak.r@gmail.com
(Received 3 April 2019; Revised 14 April 2019; Accepted 24 April 2019; Available online 3 May 2019)

***Abstract* - Multi agent is that the broader space that is developing at a fast pace towards analysis and development of distinct vary of topics. These areas agitate numerous methodologies towards the agent style and comprehensive classification theme. During this paper we tend toestablish major mode aspects of software package agents, then provides an summary of existing ontologies, and combines the most effective aspects of those themes to propose a brand new classification scheme. So as an instance the classifications, the JACK Intelligent Agents design is delineate within the context of the theme.**
***Keywords:* Agent, Cluster, Mining, Classification**

I. INTRODUCTION

Agent technology is considered to be the increasing space that comprises of several disparate areas of analysis, and offers many distinct approaches. The ascent of this field within the past decade has occurred in parallel with the evolution of the day to data analysis. Multi-agent systems will exploitmassively distributed systems over the present day web. Despite the recognition of multi-agent systems, there's little or no agreement regarding what specifically constitutes a software package agent [1]. Research in the field of agent-based systems is quite numerous [13], for a software package agent. Many approaches have been evolved for implementing agent styles and shares many key options over all approaches to agents. The basic feature of software package agent is autonomy. Similar to a human being the software package agents should be always ready to act on behalf of another party or third party may be an individual or another agent. Hence, agents should be ready to take action as needed in the framework [3].

Software package agents should run unceasingly. In contrast to abundant standard software package performs a hard and fast task then terminates, because agents run perpetually. This permits agents to watch the present scenario and take acceptable action once needed. Agents conjointly possess social ability and the power to move with alternative agents. The blessings of software package agents come back not from individual agents, however from communities of interacting agents. Variety of existing surveys and classifications of software package agents are conferred antecedently [2].

This paper develop a comprehensive classification that consists of broad selection software package agents that associate degree introduction to the essential ideas of software package agents.

II. EXISTING CLASSIFICATION SCHEMAS

Manyclassification schemes exist when dealt with software package agents, by specializing either a specific domain or on a selected variety of agent. These taxonomies are going to be explained using a brand new classification theme that combines the effective aspects. Acomprehensive classification of agents has been conferred by Nwana [1] wherever the identification of "agent" as a meta-term covers a variety of agents. The primary attributes of agents ought to exhibit the properties such as learning and cooperation,butthese are not planned as being necessary. The classes of agents are outlined such as static versus mobile and thoughtful versus that reactprimary class refers to the flexibility of agents around a network.

Taxonomy of agents over the targeted industrial applications that produce which have planned by Parunak [2]. Where the individual agents consist of factors where the operations are re-implemented. The property primarily identifies whether or not agents inside a system are based on identical design having architectures. However standard communication mechanism property identifies every agent that uses reasoning and reacting to itsatmosphere that ranges from pure reaction to pure coming up.

Another classification is conferred by Franklin and Graesser [3] where all the agents are considered to be reactive, autonomous, goal-oriented and temporary continuous with subsequent attributes being optional: communicative, learning, mobile, versatile (non-scripted) and character (personality). From these properties "natural sort taxonomy of agents" is conferred.

Wooldridge and Jennings [4] have made another classification of software system agents with weak notion of agency that is planned for shaping the associate of an agent as hardware or computer code with subsequent properties such as autonomy or social ability or reactivity or pro-activeness. One notable facet of this definition is that it needs associate in agent to each with exhibit purposeful behavior which is reactive and proactive. This is often in distinction to many alternative classifications listed on top of contemplating strictly reactive and strictly goal-oriented systems inside their classifications.

III. PROPOSED CLASSIFICATION TECHNIQUE

In this section, the classification of schemes will considerthe assorted sub-fields of software system agents. The core attributes of every software system agent comprises of properties such as: autonomy, time based continuity and social ability. Agents should be in a state to run severally with very little or no human intervention. Temporal continuity is needed as agents should run ceaselessly instead of merely performing a task, send notifications and terminate or can create multiple agents too.

Agents should comprises of social ability where software system agents return individual items of software system acts in isolation over communities of interacting agents. Additionally to the core attributes, agents could also be classified in keeping with the subsequent features

1. Pro-activeness
2. Adaptiveness Mobility
3. Collaboration
4. Veracity
5. Disposition

Each of those options could also be any sub-divided into an inventory of properties, as explained below in JACKTM Intelligent Agents are classified thoroughly.

A. JACK Intelligent Agents

"JACK Intelligent Agents (JACK)" [5] was developed using Java artificial language. It provides extension to implement agent behavior [6]. Agents have been designed with the JACK development setting square measure compiled to plain Java code. JACK relies upon the "Idea need Intention (INI)" model of computer science, it provides a high degree of autonomy and pro-activeness. Associate in agent guide showing a number of the extensions that JACK provides to Java is shown in Fig. 1.

```
agent AgentType extends Agent [implements
InterfaceName]
{
  // Knowledge bases used by the agent
  // are declared here.
  #private data BeliefType
      belief_name(arg_list);

  // Events handled, posted and sent
  // by the agent are declared here.
  #handles event EventType;
  #posts event EventType reference;
  #sends event EventType reference;

  // Plans used by the agent are
  // declared here. Order is important
  #uses plan PlanType;

  // Capabilities that the agent has
  // are declared here.
  #has capability CapabilityType
    reference;

  // other Data Member and Method
  // definitions
}
```

Fig. 1 JACK agent example

B. Pro-Activeness

An agent's pro-activeness refers to however it reacts to and reasons concerning its setting and the way it pursues towards its goals. Considering that the aim of Associate agent is to autonomously and unceasingly perform a given set of tasks on behalf of a requester, the approach that the agent takes toward achieving goals is central to its performance. Associate in agent's proactiveness could also be characterized together of the following:

1. Pure Reaction

Pure Reaction is that the simplest type of behavior and involves directly reacting to stimuli within the agent's setby mapping input from their sensors on to associate in action. This approach has principal benefits such as the agents will react quickly to external events and it greatly simplifies the method of planning agents. Despite the apparent simplicity of reactive agents the complicated behaviors will evolve from interaction with complicated environments. Brooks' devised a reactive framework referred to as the subsumption design [8] for physical robots that uses layers of reactive management systems to attain complicated behavior.

2. Pure Planning

In this process the agents will takethe goal oriented approach to attain their goals. This approach depends upon utilizing various techniques from ancient AI to spot tasks that requires to be performed so as to satisfy the goals of the agent. This approach permits flexibility within the pursuit of goals however is usually slow. The dominant technique for purposive agent behavior is that the "Belief Desire Intention" (BDI) model [9].

3. Hybrid

Hybrid agents mix the above two techniques by incorporating each reactive and proactive that provides the speedy response of reactive agents with the subtle reasoning of designing agents because of superior approach to agent style. Thenotion of agency given by Wooldridge and Jennings [4] identifies each reactivity and pro-activeness as necessary attributes of all agents. Parunak [2] divides hybrid agents into classes: "reaction overridden by plan", wherever the planner might rule the reactive elements based on the degrees and "reaction changed by plan" wherever the planner will reconfigure the reactive element to behave otherwise within the future. The latter technique needs a degree of adaptiveness inside the agent.

4. JACK

JACK offers a hybrid approach, where the traditional events correspond to a reactive approach that performs triggering a direct response. BDI events based on active (goal-directed) approachwill perform reasoning concerning arrange choice. Proactiveness is one amongst the most options of the JACK

agent system that are designed heavily based upon the BDI model that facilitatesmany options where the JACK framework provides sturdy tools for arrange specification and choice.

C. Adaptiveness

Adaptiveness describes an agent's ability to change their behavior over time which is often considered to be a key attribute when is related to agents. The keyword "agent" is commonly considered or represented to be "intelligent agents". Many AI techniques exist that helps autonomously over activity tasks. Adaptiveness is the major aspect of pro-activeness where hybrid systems will look forward to adapt this technology. Despite thisall the agents will measure adaptability without exceeding the restricted manner. Adaptiveness comprise of many different classes as noted below

1. Learning

Learning agents have the capability power to change their behavior over a specific period of time so as to adapt their practicality to their atmosphere and to enhance their effectiveness. A large vary of techniques are applied to implement learning process of agents together with memory-based learning or reinforcement learning [10] and Bayesian belief networks [11] are to be learned.

2. Subsumption

The Subsumption design will permit a designer feature a bit further "layers of competence" over a specific time period. This property was first outlined by Brooks [8] who was implemented on autonomous robots. Though they have proposed the mechanism that tailors the software package agents [12] for a specific time period. The agent designer will thus expand and adapt the agent's for a sequential iteration of development. This differs from agent learning and adaptation is performed by designers expressly adding practicality to the agent.

3. Non-Adaptive

Non-adaptive agents will measure and modify the agent behavior over time. As noted by Wooldridge [13], though the discipline of "intelligent agents".Formost part grew,though not all agents got to be capable of learning. The sole intelligence needed by agents is capable to create freelance choices that tend to act autonomously. Learning is commonly considered to be acceptable technique for agents to use. The quality can rely upon the circumstances within which the agent is being employed or deployed. In mission important applications as an example of adaptiveness could also be liable because it may lead to unpredictable behavior by the system which is non-adaptive.

4. Constraint Based

Constraint-based agents place restrictions over the capabilities of agent's who can easily adapt, so as to mitigate the issues related to learning agents over important systems. This permits several of the advantages of learning agents, whereas providing safeguards to confirm that the agent still fulfills important functions. JACK [5] development atmosphere provides some support for adaptiveness in refinement of arrange choice, learning within the ancient sense isn't a key foundation of the design. Developers could use advanced learning techniques in developing individual agents.

D. Mobility

Software agents are well-suited to tasks that need portability and movement over large-distributed networks such as web. Consequently, a lot of analysis is required into agents that had rotated around the conception of quality of an agent through:

1. Physically Mobile

Physically mobile agents are capable of transporting their executionbetween machines over a network. This provides a lovely mechanism for developing computer code for distributed systems. Qualityis commonly enforced during a clear manner which permits an agent to continue traditional execution because it travels round the network. An example of this approach is implemented in the research done by Concordia agent [14] developed by Mitsubishi electrical ITA.

2. Logically Mobile

Logically mobile agents are those which tend to physically execute on one machine, however access different logical locations through a network association. These agents could also be thought of visiting these locations through an abstract sense.Thoughthe actual execution is fastened by implanting a physical machine over mobile agents who tend to offer an acceptable mechanism for gathering knowledge from web. A typical example is a net spider agent [15] that visits and processes a series of websites by following machine readable text links called as net crawlers.

3. Static

Static agents are those who will not offer a mechanism for obtaining quality. Notlike different agent systems like IBM's Aglets [16]. Whereas JACK [5] Intelligent Agents don't offer supportfor quality by implementing it in Java-basedagents that victimizes by the aspect of Java's cross platform capabilities that supportsto publish and helps in implementing physical quality. Virtual quality may additionally be enforced at the level of agent due to no existence of field of study which is supportedby intervals of JACK system.

E. Collaboration

Collaboration among agents underpins the success of various operations or actionsimplemented in timely manner.

For this reason, agents ought to possess some variety of social ability which is divided into various types.

1. Communicative

Communicative agents are those agents who communicate with people or end users. Whoare ready to coordinate with different agents by causing and receiving messages victimization using agent communication language. This permits a high degree of collaboration with social activities like distributed downside determination and negotiation being attainable. Many agent communication languages are available in the market where the foremost distinguished being is KQML [17].

2. Non-Communicative

Non-communicative agents are those agents or people who don't have interaction in formal communication. Though direct agent communication is fascinating in several things.It isattainable for agents to collaborate while not actual communication is going down due to the interaction of agents for sharing of resources will cause cooperative or competitive raising behavior. TheSWARM [18] agent system provides a framework which communicate with the agents for a specific interval. The SWARM system is usually used for simulating social systems where plenty of such simulations demonstrate cooperative behavior towards direct communication. Aneasy example of this is often the HeatBugs [19] model.

3. JACK

Intelligent Agents provide a message-based communication framework [6] which permits a message to be passed over different named agentsfor implementing the communications mechanism. There exist many extensions to JACK square measure that offers an additional advanced support for performing communication and to collaborate with more than one agent. JACK groups [20] permit agents to be classified into groups and sub-teams that act as separate reasoning entities to supportcomplete BDI model, with the existence of agent based beliefs or desires or intentions or team-level plans. FIPA JACK [21] and extension to JACK is built on the top of JACK's whose basic communication framework will supply a FIPA compliant communications infrastructure.

F. Veracity

Collaborating agents may be communicative or non-communicative based on their origin properties due to which an agent might conceive to deceive different agents via their messages or behavior or notifications. Agents might thence be classified by their veracity:

1. Truthful

Truthful agents are those who don't conceive to deliberately deceive other agents. In exceedingly closed surroundingsthe truthfulness of all the agents is considered to be secured based on theirsimplicity of negotiation and interaction process. If associate degree of an agent indicates that it will provide a service over different agents, will assume to provide a trial to supply build. If associate degree agent provides data to help with a satisfying goal the different agents may be fairly sure whether the data is correct or not.

2. Untruthful

Lying agents square measure those who might conceive to deceive different agents either by providing false data or by acting in an exceedingly deceptive manner. Wherevernumerous agents from completely different vendors contend to attain their own goals the problem of deception becomes an element. JACK is meant primarily to be used in closed environments to obtain the overall goal. FIPA JACK permits it to be employed in open environments like the Internet which interacts with agents designed to exploitdifferent architectures and methodologies. In such a scenario a designer might have to require into consideration the chance of lying agents.

G. Disposition

The final consider this classification of agents indicates the "attitude" of the agent toward different agents, and its temperament to get together with them. Agents could also be classified as:

1. Benevolent

Benevolent agents will continuously conceive to perform a task once it's receives a request. Wooldridge [13] identifies benevolence as "the assumption that agents share common goals" which are very similar to truthful agents. The process of collaboration is greatly simplified in an exceedingly system that consists of entirely benevolent agents.

2. Self-Interested

Self-interested agents will act on their own interest, collaborating with different agents only is considered to be helpful to try. These types of agents are not expected to try the process of coordinating with other agents which is a tough task. In several cases, self-interested agents could also be competitive when compared with other agents to attain a predefined goal or acting to secure a stronger deal than different agents. A typical example of this is to associate degree of electronic auction system.

3. Malevolent

Malevolent agents are those who commit to inconvenience the other agents, or undermine them in a way. Unlike self-interested agents who tend to merely act to realize their own goals rather than they act in some predefined malicious manner. While the distinction between self-interested agents and malevolent ones might merely be considered to be

distinction within the goals of the agent.Self-interested agents might have a positive impact over the system, where the presence of malevolent agents inside a system is unlikely to be of any profit. Sometimes a worm could also be considered to be a kind of malevolent agent.

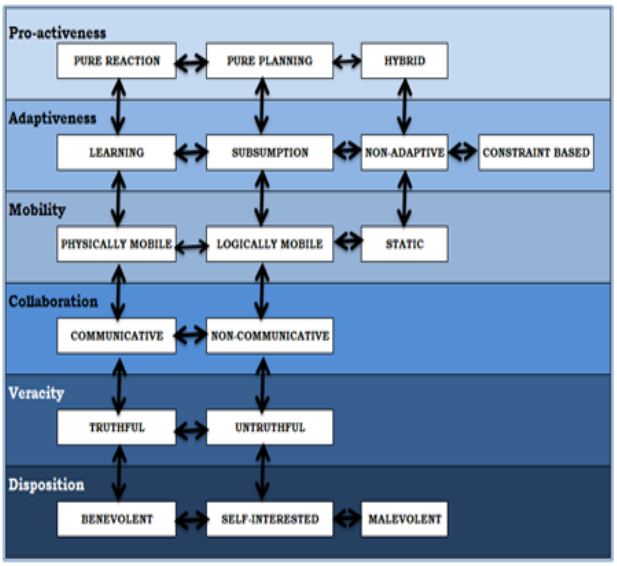

Fig. 2 Classification of agents

IV. CONCLUSION AND FUTURE SCOPE

This paper describes a replacement classification theme for Agent Technology. It draws upon many existing ontologies, however provides associate degree all inclusive classification that takes under consideration the assorted aspects of agent analysis.

One grouping for mobile agents is insufficient since nearly any combination of the on top of properties is possible. Additionally agent architectures could also be neutral regarding sure classifications that conferred during this paper permits all manner of agents to be enclosed inside the theme.

Apart from merely providing a mechanism to analyze and catalogue agents this classification provides a technique that determines well varied agents who can act inside a system. Agents that share an oversized range of options are going to be higher ready to collaborate; communicative agents can perform far better in associate degree setting with alternative agents that share this attribute whereas static agents can do poorly in associate degree setting designed for mobile agents. While this paper attempts associate degree tries to outline an inclusive classification technique for agent-based systems which is no means that definitive. More work must be done on processing the classes, associate degreed reaching accord on metaphysics for agent-based systems.

REFERENCES

[1] H. S. Nwana, "Software Agents: An Overview", *Knowledge Engineering Review*, Vol. 11, pp. 205-224, October/November 1996.
[2] H. V. D. Paranuk, "Applications of Distributed Artificial Intelligence in Industry",*Foundations of Distributed Artificial Intelligence*, 1996, pp. 139-164.
[3] S. Franklin, and A. Graesser, "Is it an Agent, or just a Program: A Taxonomy for Autonomous Agents", *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag, 1996.
[4] M. J. Wooldridge, and N. R. Jennings, "Intelligent Agents: Theory and Practice",*Knowledge Engineering Review*, Vol. 10, pp. 115-152, Jun 1995.
[5] R. A. Brooks, "A Robust Layered Control System For A Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. RA-2, pp. 14-23, March 1986.
[6] A. S. Rao, and M. P. Georgeff, "BDI agents: from theory to practice", *Proc. First Intl.Conference on Multiagent Systems*, San Francisco, 1995.
[7] R. Kozierok, and P. Maes, "A Learning Interface Agent for Scheduling Meetings", *Proc. 1st International Conference on Intelligent User Interfaces*, pp. 81-88, 1993.
[8] Yi~Shang, H. Shi, and S.S. Chen, "An Intelligent Distributed Environment for Active Learning", in *Proc. ACM Journal of Educational Resources in Computing*, 2001.
[9] H. Song, S. Franklin, and A. Negatu, "SUMPY: A Fuzzy Software Agent",*proc. ISCA Conference on Intelligent Systems*, 1996.
[10] M. Wooldridge, "Agent-Based Software Engineering", *IEE Proceedings Software Engineering*, Vol. 144, pp. 26-37, 1997.
[11] C. Cesarano, A. d'Acierno, and A. Picariello, "An intelligent search agent system for semantic information retrieval on the internet", *Proc. fifth ACM international workshop on Web information and data management*, pp. 111-117, 2003.
[12] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an Agent Communication Language", *Proc. International Conference on Information and Knowledge Management*, 1994.
[13] P. Praveen, and B. Rama, "A Novel Approach to Improve the Performance of Divisive Clustering-BST", *Data Engineering and Intelligent Computing. Advances in Intelligent Systems and Computing*, Vol 542. Springer, Singapore.
[14] P. Praveen, C. J. Babu, and B. Rama, "Big data environment for geospatial data analysis", *Proc. 2016 International Conference on Communication and Electronics Systems (ICCES)*, pp. 1-6, Coimbatore, 2016.
[15] P. Praveen, and B. Rama, "An empirical comparison of Clustering using hierarchical methods and K-means", *Proc. 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 445-449, Chennai, 2016.
[16] P. Praveen, B. Rama, and T. Sampath Kumar, "An efficient clustering algorithm of minimum Spanning Tree", *Proc. 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 131-135.Chennai, 2017.
[17] R. Ravi Kumar, M. Babu Reddy and P. Praveen, "A review of feature subset selection on unsupervised learning", *Proc. 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 163-167, Chennai, 2017.
[18] Pappula, Praveen, and Rama B. Ramesh Javvaji, "Experimental Survey on Data Mining Techniques for Association rule mining", *International Journal of Advanced Research in Computer Science and Software Engineering*, 2014.