WAIMS© World Academy of Informatics and Management Sciences, Vol1 Issue 2, April-May 2012
ISSN (online):2278-1315
www.waims.co.in

MULTI AGENT FRAMEWORK FOR MEASURING SOFTWARE RELIABILITY

[1]Reema Ajmera, [2]Dr. C.S Lamba

[1]Resarch Scholar, NIMS University Rajasthan Jaipur
[2] Research Supervisor, NIMS University, Jaipur

Abstract:

Agent Oriented modelling in software engineering is a new phenomena, we have observed that Agent-mediated environoment and their technique represent an exciting new means of analysing, designing and building complex software systems. Agent oriented software engineering significantly improves current practice in software engineering and to extend the assortment of applications that can be tackled fesibly . Yet, to date, , no standard methodologies, development tools, or software architectures. In this paper we are emphasized on Agent-oriented modeling approach to the conceptualization process. It defines agent models and proposes a high-level methodology for agent oriented analysis and design. Agent-oriented approaches can significantly enhance our ability to model, design and build complex (distributed) software systems. This paper therefore provides agent theory for software scenario.

Key words: Multi Agent

I,INTRODUCTION

Agent-based computing embodies a novel software engineering paradigm that has emanate merging of two technologies i.e. artificial intelligence (AI) and object-oriented distributed computing. The goal of Agent-based systems is to wallop a balance between artificial intelligence and computational utility. Agents are intelligent, autonomous, software components capable of interacting with others within an application, attaining a common goal and thereby contributing to the resolution of some given problem. Multi-agent systems (MAS) are composed of a set of agents and are useful for the modeling and development of distributed information systems with synchronous or asynchronous component interactions. Multi-agent systems differ from non-agent-based systems because agents are intended to be autonomous units of intelligent functionality who can interact with others through high-level protocols and languages. A system can be successfully built and deployed, if it has been properly conceptualized. Conceptualization requires an appropriate set of abstractions and a methodology for system specification, analysis, and design. Software agents offer greater flexibility and adaptability than traditional components. Agent-oriented software engineering allows developers to use a set of high-level, flexible abstractions to represent and understand systems [1]. However, many of our traditional ways of thinking about and designing software do not fit the multi-agent paradigm [2] and Rapid integration of distributed agents provides opportunities to build such software systems. [1] general rule for agent within MAS is “Keep It Small and Simple” (KISS).

In this paper, we introduce an agent-oriented paradigm for conceptualizing the analysis and design of agent-based systems. The remainder of the paper is organized as follows. In Section 2, we take a more detailed look at intelligent agents, their characteristics, capabilities and interactions. In Section 3, we introduce the agent oriented framework and describe the steps to get the goal for a defined scenario. Finally, in Section 4, we summarize our results and provide an insight to the future challenges.

II. AGENT ONTOLOGY

An agent is an assistant that works on behalf of others (agents or humans). In its simplest form, an agent represents an actor and is thereby an embodiment of its actions. In AI, an intelligent agent refers to a software component that lives inside computer environments, operates continuously and autonomously and cooperates with similar entities. An agent is associated with its mental state that can be composed of components like belief, capabilities, choices and commitments. In [4], Wooldridge and Jennings have introduced weak and strong notions of agencies. They have used the stronger notion of an agency to imply ‘‘a computer system that, in addition to having some basic properties, can be either conceptualized or implemented using concepts that are more usually applied to humans. [1]

A.AGENT CHARACTERSTICS

A characteristic is an intrinsic or physical property of an agent. The following are some common agent characteristics [3,4].

Autonomy: An agent can act on another's behalf without much guidance.

Communication: An agent can communicate with other agents on a common topic of discourse by exchanging a sequence of messages in a speech-act based language that others understand. The domain of discourse is described by its ontology.

Mobility: An agent can migrate from one system to another in a pre-determined fashion or at its own discretion. Accordingly, agents can be static or mobile.

Learning: An agent can have the ability to learn new information about the environment in which it is deployed and dynamically improve upon its own behavior.

Cooperation: An agent can collaborate and cooperate with other agents or its user, during its execution, to minimize redundancy and to solve a common problem.

Of the five, the first two are considered by many to be the most basic agent characteristics, while the second and third together contribute towards another paradigm for distributed computing (also known as agent-oriented programming).

B. AGENT COMMUNICATION

Agents are entities that are able to respond to the environment. It simply sends a messgae to communicate with another agent which is targeted destination. On the other end receiver agent have potential to select what type of message it accepts and what type is dened. Another way of choosing with whom an agent wishes to communicate is messages prioritizing. It also allows the receiver to control its own behavior in order not to get overwhelmed by messages if it has other important work to do. The messages can also have many parameters – for example in the human world a gossip with time loses its values and so does also in the agent world. Also filtering receivers is important – for example an agent's domain is football, so it wants to communicate with other agents that are interested in football only. The reason why messages are so important is because this is the basics of communication. An intelligent agent is the one, who can show his needs to the others and cooperate, but what is also important – that other being must understand what one wants from them, so the common language must be defined. It is very similar to humans that two people from different countries cannot communicate verbally when they speak different languages.

C.AGENT INTERACTION

When trying to solve problems using agents, it soon becomes clear, that there has to be more than one agent. Moreover, they have decentralized nature of the problem. Sometimes these views can be so different, that agents can compete with each other, because competition many times brings the higher quality of products and services. To solve the problem agents have to interact to achieve individual goals and so sometimes compete with others. This interaction may sometimes bring relationships like trust, antagonism, and many other human-like behaviors. Sometimes it involves also cooperation abilities, acting in the name of others or also forming communities in order to achieve the common goal. In majority of the cases agent are more selfish. That means they are trying to achieve their own goal without looking at others, but when they cooperate, there often form structures that characterize the cooperating group. Like in the army there might be some leaders responsible and also commanding a set of agents. But to form that kind of relationship, they have to have the abilities defined before. Relationships, like those between humans, can change in time, which is a very important element and they can be weakened or evolve to a permanent bond.

Whatever is the nature of the social interactions; there are two points that differentiate them from those that exist in computational models [5]. The first one is that agent-oriented interactions seem to be more complicated than in other contexts. Those involve: negotiations, cooperation and coordination. The second one is that agents have limited control over their environment and limited possibilities of observing it, so they have to solve problems flexibly. Their interactions have to be handled in a flexible way and agents need already defined responses to the events in the life cycle, but they also need the ability to makes decisions when unexpected situations occur. In order to make this possible, there is a set of techniques applied [5] like reinforcement learning, mechanisms design and electronic institutions.

III. AGENT ORIRNTED FRAMEWORK

A Multi Agent Framework for a Software scenario (Figure 1) proposed a set of services that can be requested either by users or by other external component. The framework is an environment composed of a Master agent and one or several of Service agents. The services provided by a framework are performed by different agents assigned by master

agent. Service agents are having different functionalities which are specific to the scenario and characterized by the information systems.

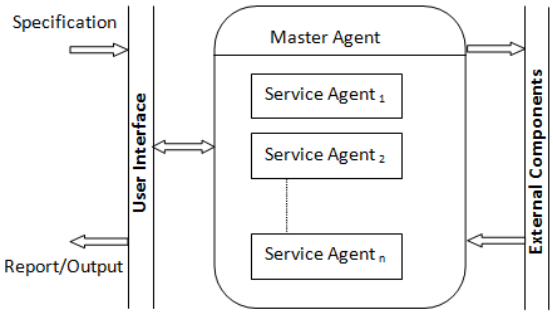

Figure.1.Multi Agent Framework for a Software Scenario

The idea behind agent-oriented thinking is to explore the applicability of the agent paradigm in conceptualizing a problem domain. In a very broad sense, it is about visualizing actors interacting with others in their environments for solving the given problem. An actor will act, based on its perceptions. It may also realize that it has insufficient knowledge and seek that supplementary information elsewhere to conduct an activity. Actors, activities and their coordination will lead to the conception of scenarios, which will get stitched together as the thinking process gradually evolves towards the goal. The key concepts in this thinking process are the actor, activity and scenario. Accordingly, we define the following.

Master Agent: An entity that performs some actions (fundamental units of behavior) based on its state (perceptions) and can interact with other entities (active or inactive) in the problem domain under consideration. An actor can play one or more roles.[1] Master agent performs certain operations that are :

Identifies service agent

Assign task to service agents

Cumulated results/ report received from Services Agents

Service Agent: Performs activity that is composed of one or more actions to hit the goal, it provides a context for its constituent actions. An activity can be followed by another activity. Activities performed by service agents are:

Each Service agent perform assigned task

Service agents are capable to interact with other agent through master agent.

Each agent acquiesce report to master agent

Agent-oriented analysis

We define Agent-oriented analysis (AOA) as the process that includes all activities (steps) for discovering, documenting and maintaining a set of specifications for actors (roles), activities (responsibilities) and interactions for the design of an agent based system. It may then primarily consist of the following steps:

1. Study the scenario description and identify all the possible actors and main activities

2. Every actor, roles are determined. It may be worth noting here that an actor may play multiple roles in the same or different contexts. For each actor and each role, enlist all the possible responsibilities, based on the role definition and activities associated with the actor. Actors, roles and responsibilities will help design the agent classes.

3. For each identified activity, analyze the context (causes and consequences) to determine all interactions (if any). It may be worthwhile to note here that agent interactions may often fall into typical patterns of behavior that have been previously identified and used in other applications. So the process may begin by looking for these patterns. For each distinct interaction, determine the actors, exchange details, the typical pattern of behavior, the speech-acts used for the communication and the associated knowledge. This will help in designing the mental states of the agents and the ontology later.

4. Refine the analysis. For each agent and interaction, remove duplication and redundant information.

In given framework each agent performs their assigned activities and responsibilities.

IV. CONCLUSIONS

In this paper we explore the agent and their characteristics and in proceeding of our research work we will proposed a software testing framework using the concepts of Multi Agent..

V. REFERENCES

[1] Pratik K. Biswas, Towards an agent-oriented approach to conceptualization, SciencDirect, Applied Soft Computing 8 (2008) 127–139

[2] Martin L.Griss et.al,Accelerating Development with Agent Components,Software Engineering Internet time

[3] P. Morreale, Agents on the move, IEEE Spectr. (1998).

[4] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowl. Eng. Rev. 10 (2) (1995) 115–152.

[5] M. Elammari, W. Lalonde, An agent-oriented methodology: high-level view and intermediate models, in: Proc. of AOIS Workshop, Seattle, WA, 1999.

VI AUTHORS PROFILE

Dr. C. S Lamba has PhD degree in Computer science and currently working as a Head of department, RIET Jaipur Rajasthan India. He has more than 14 years of experience and holding diversified knowledge in the field of Information technology. He has published and Participated research paper / article in various conference/ seminar globally

Reema Ajmera received her M.Tech (C.S.) degree from Banasthali Vidhypeeph in 2007. She has some important books as a author as well as currently she is associating with the Jaipur National University at department of computer and systems sciences. She has also attended conference and seminar at repute