A Structural Analysis of Software Agent in Terms of its Behavior and its Interaction With System

Anand Kumar Pandey¹, Rashmi Pandey² and Deependra Gautam³

¹Asst. Prof. ITM University, Gwalior
² Asst. Prof. ITM GOI, Gwalior
³Asst. Prof. MPCT, Gwalior

Abstract

The term software agent is related to the usage of a computational model in order to improve the understanding of a system's behavior and/or to evaluate strategies for its operation, in explanatory or predictive schemes. Agent-Based models and Multi-Agent Systems (MAS) have been adopted to simulate very different kinds of complex systems, from the simulation of socio-economic systems to the elaboration of scenarios for logistics optimization, from biological systems to urban planning. We propose a preliminary methodology for agent-oriented software engineering based on the idea of agent interaction analysis. This approach uses interactions between undetermined agents as the primary component of analysis and design. This paper discusses the specific aspects of this approach to modeling and simulation from the perspective of Informatics, describing the typical elements of an agent-based simulation model and the relevant research.

Keywords: Agent Based Model (ABM), Autonomous, Reactive Agents, Agent Communication Language (ACL).

1. Introduction

A software agent is a computer program that is designed to assist a user both directly and indirectly, depending on how it is used. Software agents can be installed on a computer to perform a specific task or to alert a user for a specific action. Developing software application is a science under continuous development to keep track with the evolutions in programming paradigms, computer resources and needs, for a more “clear” and adequate representation for software systems developing process. This concept is related to software engineering which is about to design and develop high quality software. The generic software engineering life cycle begins from requirements and specification analysis, system design and program coding, until it reaches system testing and delivery. The agent paradigm has, over recent years, given rise to a large amount of research on the internal structure of agents as general problem solvers capable of effective intelligent behavior in dynamic environments.

A software agent is autonomous or at least semi-autonomous, meaning it can perform tasks in pursuit of a goal with minimum of no direct supervision or direct control, but can interact with another entity to obtain guidance or output results [3]. It may or may not have a user interface. The full potential of agent-based systems in solving problems in complex domains depends upon the systems themselves, and the designs from which they are constructed. An abstraction of the social aspects of an agent can be given as a system role, and this concept is used in many of the emerging agent-oriented methodologies. Many situations and subsystems can be viewed as being characterized by the presence of a number of autonomous entities whose behaviors (actions and interactions) determine (in a non-trivial way) the evolution of the overall system. Agent-based models are particularly suited to tackle these situations and they support the study and analysis of topics like decentralized decision making, local-global interactions, self-organization, emergence and effects of heterogeneity in the organized system.

2. Agent Behavior Specification

The specification of agents' behaviors can be considered as being composed of two elements. First of all, a proper representation of agent actions must be given. Actions are the elements at the basis of agent behavior: they can cause modifications in their environment or in other agents that constitutes the ABM [1]. Different modeling solutions can be provided in order to describe agents' actions, such as the following:

- The transformation of the global state of the system;
- A local modification of the environment;
- A response to influences associated to agent perceptions;
- The execution of computing processes, carried out internally by an agent to elaborate perceived data and change its own state;
- A physical displacement of an agent in the spatial structure of the environment.

A general schema describing the role and interaction of software system with the systems as a predictive or explanatory instrument is shown in Figure 1.
The agents’ behavioral specification, however, also includes the mechanisms effectively selecting the actions to be carried out, according to the perceptions and internal state of an agent. The term architecture refers to the model of agent internal structure that is responsible of action selection. Different architectures have been proposed in order to obtain specific agent behaviors and they are generally classified into deliberative and reactive. Deliberative or cognitive agents, instead, are characterized by a more complex action selection mechanism, and their behavior is based on so called mental states, on facts representing agent knowledge about the environment and possibly, also on memories of past experiences. Deliberative agents, for every possible sequence of perceptions, try to select a sequence of actions allowing them to achieve a given goal [2]. Reactive agents are elementary (and often memory less) agents with a defined position in the environment. Reactive agents perform their actions as a consequence of the perception of stimuli coming either from other agents or from the current environment.

3. Agent Interaction

Interaction is a key aspect in Agent based model, there is a variety of definitions for the concept of agent and most of them emphasize the fact that this kind of entity should be able to interact with their environment and with other entities in order to solve problems or simply reach their goals according to coordination, cooperation or competition schemes [6]. We explore a different approach, named agent interaction analysis, based on using the interactions between agents as a primary component of analysis [4]. As with the interaction methodology, this approach does not constrain the organization or role model until late in the analysis, after domain-specific requirements have been identified. Agent interaction analysis, by translating requirements into agent interactions rather than roles, also allows for the system behavior to be flexibly distributed between the designed system and connected systems [8].

This section presents a conceptual taxonomy of currently known/available agent interaction models (see Figure 2), trying to define advantages and issues related to them, both from a conceptual and a technical point of view.

From a strictly technical point of view, agent communication is generally indirect even in direct agent interaction models [5]. In fact, most of these approaches adopt some kind of communication infrastructure, supplying a reliable end-to-end message passing mechanism. The first and most widely adopted kind of an agent interaction model provides a direct information exchange between communication partners [7]. This approach ignores the details related to the communication channel that allows the interaction, and does not include it as an element of the abstract interaction model.

4. Conclusion:

Intuitively an Agent Communication Language (ACL) provides agents with a means of exchanging information and knowledge (see Figure 4). This vague definition inherently includes the aspect of the conception of the term agent, which assumes that an agent is an intelligent autonomous entity whose features include some sort of social ability. In this paper the proposed framework often provides abstractions and mechanisms for the definition of agents and their environments to support their interaction with other systems.
References:


AUTHOR

[1] Anand K. Pandey, MCA, M.Tech., pursuing Ph.D. from Mewad University, Chittorgar, Rajasthan, is currently working as a Asst. Prof. in School of Computer application in ITM University, Gwalior, has been published 7 papers in referred journals and 10 paper at national and international conferences. He also written a book titled “Simulation and Modeling” for BE, MCA and M.Tech, students according to RGPV University.

[2] Rashmi Pandey, MCA, is currently working as a Asst. Prof. of computer application dept in Institute of Technology and Management (ITM), Gwalior, has been published 6 papers in referred journals, and 5 papers in national and international conferences. She also written a book titled “Active Server pages” for B.C.A., BE, MCA and M.Tech. students according to Jiwaji and RGPV University.

[3] Deependra Gautam, MCA, is currently working as a Asst. Prof. of computer application dept in MPCT, Gwalior. He is pursuing M.Tech in C.S.