Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 8 (2008) 127–139
Applied Soft Computing
www.elsevier.com/locate/asoc

Towards an agent-oriented approach to conceptualization

Pratik K. Biswas
Avaya Inc., 307 Middletown-Lincroft Road, Lincroft, NJ 07738, USA

Received 3 January 2006; received in revised form 27 October 2006; accepted 22 November 2006
Available online 10 January 2007

Abstract

Agent-oriented modeling provides a new technique for the conceptualization of agent-based systems. This paper extends and formalizes this agent-oriented modeling approach to the conceptualization process. It defines agent models and proposes a high-level methodology for agent-oriented analysis and design. It also includes analogies with the object-oriented and other existing agent-oriented methodologies, wherever applicable. The paper is concluded with a case study and an insight to future challenges.

Keywords: Intelligent agents; Multi-agent systems; Agent-oriented software engineering; Agent-oriented thinking; Agent-oriented modeling; Extended agent model; Agent-oriented analysis; Agent-oriented design

1. Introduction

Agent-based computing represents a novel *software engineering paradigm* that has emerged from merging two technologies [1], namely, artificial intelligence (AI) and object-oriented distributed computing. *Agent-based systems* aim to strike 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. They are beneficial, as they can exist within such important applications like electronic commerce and information retrieval. *Multi-agent systems* (MAS) [4,5,10–12] 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. However, many of our traditional ways of thinking about and designing software do not fit the multi-agent paradigm. In this paper, we advocate 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 concept of *agent-oriented thinking*. In Section 4, we discuss *agent-oriented modeling*, define a new agent model, describe the steps involved in our approaches to *agent-oriented analysis* and *design*, and compare them with *object-oriented* methodologies. In Section 5, we present a case study to illustrate the applicability of our methodology. In Section 6, we compare our methodology with other existing *agent-oriented* methodologies. Finally, in Section 7, we summarize our results and provide an insight to the future challenges.

2. Terminologies

In this section, we explain and clarify certain terminologies that are closely associated with *agent-oriented software engineering* (AOSE) and which we have used in describing our methodology.

2.1. Intelligent agents

Here, we discuss in detail about what do people mean by the term *intelligent agent* when they use it often in the world of AI.

E-mail address: pbiswas@avaya.com.

1568-4946/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2006.11.009

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 [7], 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."

The interactive nature of multi-agent systems calls for consensus on agent interfaces in order to support interoperability among agents coming from various sources. Foundation of intelligent physical agents (FIPA) has developed standards for building interoperable agent-based systems.

2.2. Agent characteristics

A characteristic is an intrinsic or physical property of an agent. The following are some common agent characteristics [6,7].

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*). We have used some of these characteristics to define our models.

2.3. Intelligence in intelligent agents

The *intelligence* in intelligent agents consists of its *mental state* [23] and the underlying knowledge coupled with the problem solver, which can be artificially programmed through *reasoning*, *learning*, *planning*, *searching*, *other methods or a combination of some of these techniques*. The *mental state* is sometimes captured through the *belief, desire and intention* (BDI) model [24] or other similar *mental components* like *capabilities*, *choices*, *commitments*, *goals*, etc. [23]. The underlying knowledge may or may not be part of the agent structure but is always accessible to it.

2.4. Agent capabilities

Intelligent agents have frequently been used in AI applications to communicate, negotiate, move, reason, search, match, learn, predict, adapt, repair, sense, deliver, clone, etc. These *atypical activities* often end up defining *agent capabilities*. It would be useful to remember some of these agent capabilities when we think about agents and designing agent-based applications.

2.5. Interaction protocols

Agent *interactions* often fall under *typical patterns of behavior*, technically known as *interaction protocols* (IP) [29]. An agent can select a specific protocol to initiate an interaction. Typical patterns of behavior include *request–inform* (*FIPA-Request*), *request-when-inform* (*FIPA-Request-when*), *subscription*, *advertisement*, *broadcasting*, *brokering*, etc. *Brokering* may further include activities like *bidding*, *iterative bidding*, *recommending*, *recruiting*, etc. Again, these can be used as basic tools in conceptualizing agent interactions to build agent-based applications.

3. Agent-oriented thinking

Thinking about thinking is often a complicated process. But sometimes it can be useful if we can think in advance about how we are going to solve certain problems in future. This thinking process can be either *structured* or *unstructured* but seldom both simultaneously [2].

With the *structured* approach, one would start with an overall plan for the proposed solution and work downwards to sort out all the structural elements at an ever-increasing level of details. With *unstructured* thinking, one need not start with any fixed plan but might start anywhere and build up a plan for the solution from small, self-contained sub-plans that get fitted together, as one thinks forward. *Agent-oriented thinking* often falls under the second category and works well in uncertain domains.

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.