Intelligent MAS in System Engineering and Robotics

G. Nicolás Marichal
University of La Laguna, Spain

Evelio J. González
University of La Laguna, Spain

INTRODUCTION

The concept of agent has been successfully used in a wide range of applications such as Robotics, e-commerce, agent-assisted user training, military transport or health-care. The origin of this concept can be located in 1977, when Carl Hewitt proposed the idea of an interactive object called *actor*. This actor was defined as a computational agent, which has a mail address and a behaviour (Hewitt, 1977). Actors receive messages from other actors and carry out their tasks in a concurrent way.

It is difficult that a single agent could be sufficient to carry out a relatively complex task. The usual approach consists of a society of agents - called Multiagent Systems (MAS) -, which communicate and collaborate among them and they are coordinated when pursuing a goal.

The purpose of this chapter is to analyze the aspects related to the application of MAS to System Engineering and Robotics, focusing on those approaches that combine MAS with other Artificial Intelligence (AI) techniques.

BACKGROUND

There is not an academic definition accepted by every researcher about the term agent. In fact, agent researchers have offered a variety of definitions explicating his or her particular use of the word. An extensive list of these definitions can be found in (Franklin and Graesser, 1996). It does not fall in the scope of this chapter to reproduce that list. However, we will include some of them, in order to illustrate how heterogeneous these definitions are.

“Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.” (Maes, 1995, p. 108)

“Autonomous agents are systems capable of autonomous, purposeful action in the real world.” (Brustoloni, 1991, p. 265)

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.” (Russell and Norvig, 1995, p. 31)

Despite the existing plethora of definitions, agents are often characterized by only describing their features (long-live, autonomy, reactivity, proactivity, collaboration, ability to perform in a dynamic and unpredictable environment, etc.). With these characteristics, users can delegate to agents tasks designed to be carried out without human intervention, for instance, as personal assistants that learn from its user.

In most of applications, a standalone agent is not sufficient for carrying out the desired task: agents are forced to interact with other agents, forming a MAS. Due to their capacity of flexible autonomous action, MAS can treat with open – or at least highly dynamic or uncertain- environments. On the other hand, MAS can effectively manage situations where distributed systems are needed: the problem being solved is itself distributed, the data are geographically distributed, systems with many components and huge content, systems with future extensions, etc. A researcher could include a single agent to implement all the tasks. Nevertheless, this type of macroagent represents a bottleneck for the system speed, reliability and management.

It is clear that the design of a MAS is more complex than a single agent. Apart from the code for the treatment of the task-problem, a developer needs to implement those aspects related to communication, negotiation among the agents and its organization in the system. Nevertheless, it has been shown that MAS offer more than they cost (Cockburn, 1996) (Gonzalez, 2006) (Gonzalez, 2006b) (Gyurjyan, 2003) (Seilonen 2005).

MAS, AI AND SYSTEM ENGINEERING

An important topic in System Engineering is that of process control problem. We can define it as the one of manipulating the input variables of a dynamic system in an attempt to influence over the output variables in a desired fashion, for example, to achieve certain values or certain rates (Jacquot, 1981). In this context, as other Engineering disciplines, we can find a lot of relevant formalisms and standards, whose descriptions are out of the scope of this chapter. An interested reader can get an introductory presentation of these aspects in (Jacquot, 1981).

Despite their advantages, there are few approaches to the application of MAS technology to process automation (much less than applications to other fields such as manufacturing industry). Some reasons for this lack of application can be found in (Seilonen, 2005):

- Process automation requires run-time specifications that are difficult to reach by the current agent technology.
- The parameters in the automation process design are usually interconnected in a strict way, thus it is highly difficult to decompose the task into agent behaviors.
- Lack of parallelism to be modeled through agents.

In spite of these difficulties, some significant approaches to the application of MAS to process control can be distinguished:

- An interesting approach of application of MAS to process control is that in which communication techniques among agents are used as a mechanism of integration among systems independently designed. An example of this approach is the ARCHON (*Architecture for Cooperative Heterogeneous on-line systems*) architecture (Cockburn, 1996) that has been used in at least three engineering domains: Electricity Transportation, Electricity Distribution and Particle Accelerator Control. In ARCHON, each application program (known as *Intelligent System*) is provided with a layer (called *Archon Layer*) that allows it to transfer data/messages to other Intelligent Systems.
- A second approach consists of those systems that implement a closed loop-based control. In this sense, we will cite the work of (Velasco et al., 1996) for the control of a thermal central.
- A different proposal consists of complementing a pre-existing process automation system with agent technology. In other words, it is a complementation, not a replacement. The agent system is an additional layer that supervises the automation system and reconfigures it when it is necessary. Seilonen et al. also propose a specification of a BDI-model-based agent platform for process automation (Seilonen, 2005).
- V. Gyurjyan et al. (2003) propose a controller system architecture with the ability of combining heterogeneous processes and/or control systems in a homogeneous environment. This architecture (based on the FIPA standard) develops the agents as a level of abstraction and uses a description of the control system in a language called COOL (*Control Oriented Ontology Language*).
- Tetiker et al. (2006) propose a decentralized multi-layered agent structure for the control of distributed reactor networks where local control agents individually decide on their own objectives allowing the framework to achieve multiple local objectives concurrently at different parts of the network. On top of that layer, a global observer agent continuously monitors the system.
- Horling, Lesser et al. (2006) describe a soft real-time control architecture designed to address temporal and ordering constraints, shared resources and the lack of a complete and consistent world view. From challenges encountered in a real-time distributed sensor allocation environment, the system is able to generate schedules respecting temporal, structural and resource constraints, to merge new goals with existing ones, and to detect and handle unexpected results from activities. Other proposal of real-time control architecture