An Agent Model in a Multiagent System Architecture for Automating Distributed Systems

Walt Truszkowski

NASA/GSFC, Greenbelt, MD 20771
Walt.Truszkowski@gsfc.nasa.gov

Jide Odubiyi

Loral Aerospace, Seabrook, MD 20706
jideo@groucho.aerosys.loral.com

Ed Ruberton

Loral Aerospace, Seabrook, MD 20706
rube@groucho.aerosys.loral.com

In this paper we discuss the conceptual foundation for a Multi-Agent System (MAS) which is being prototyped to automate mission operations for unmanned scientific satellites. A basic agent model has been developed and is used to characterize the agents of the MAS. Using this model three classes of agents have been currently identified and incorporated into the MAS prototype. These are: the user interface agent which serves as the main point of interaction between a user and the MAS; an agent manager which is responsible for coordinating the activities of a community of agents within the MAS; and domain agents whose specialized knowledge enable them to accomplish domain-specific tasks. In the context of the MAS prototype architecture and with a current operational focus that supports automated distributed information processing, spacecraft subsystem monitoring and growth in agent population to reflect changing user needs, we plan to address several research issues such as user modeling, cooperative activity among agents, handling incomplete requests, and agent-agent and user-agent interactions.

A Basic Agent Model and Multiagent System Architecture: Each agent in MAS is a computer-based autonomous process which is capable of goal-directed activity. It can accomplish tasks delegated to it with minimal reliance on human intervention. Because it is goal-directed it can allow the user to simply specify what he/she wants, leaving the how and where to get the information or services to the agents. It is also able to participate in cooperative work as an associate with humans or as part of a community of cooperating agents. Each agent in our MAS architecture has five high-level behavior characteristics. The attributes or capabilities are migration, autonomy, spawning, persistence, and communication. Migration is the ability of an agent to relocate to other nodes to accomplish its tasks. This ability can support load balancing, improve efficiencies of communication, and provide unique services which may not be available at a local node. Autonomy/semi-autonomy is the ability to respond to a dynamic environment without human intervention, thus improving the productivity of the user. Spawning is the ability to create other agents to support the parent agent, thereby promoting dynamic parallelism and thus fault-tolerance. Persistence is the ability to recover from environmental crashes and support time-extended activities, thus reducing the need for constant polling of the agent's welfare by the user and better use of the system's communication bandwidth. Communication provides the mechanisms for supporting agent-agent and user-agent interactions. Embedded in these five high-level capabilities is the ability of each agent to learn and to respond to stimuli.