Using Learning in a Control Agent

James J. Pomykalsk

Integrated Science and Technology and Computer Science Programs
James Madison Universit
Harrisonburg, VA 22807
pomykajj@jmu.edu

Abstract

The NASA Goddard Space Flight Center has undertaken an R&D project whose near-term objective is to achieve a higher level of autonomy in ground-based system operations. The multi-agent system, named LOGOS, is designed to replace human operators in the satellite ground control centers. This paper focuses on the on-going development of the control agent for LOGOS and the use of learning technologies to ensure the long-term success of LOGOS.

Introduction

In this paper we discuss the application of learning technologies for a control agent within a multi-agent system designed to support "lights-out" operation of a spacecraft ground system center. The multi-agent project, named LOGOS, for Lights-Out Ground Operations Systems, is an on-going research and development effort being undertaken at the NASA Goddard Space Flight Center in Greenbelt, MD.

Currently, there exists a major initiative throughout NASA to utilize current and future computing technologies, like agents, to achieve a higher level of autonomy in both ground and space based systems. The Goddard effort focuses on the analysis of the role of human operators in current ground system centers A near-term objective of ground system autonomy is to achieve a complete and comprehensive realization of "lights-out" operations. Lights-out operations are just that—operation of a ground control center without the presence or direct intervention of people. Essentially, the agents "replace" in some sense the operators and assume the responsibility for their control center activities. These activities include:

1. Planning & scheduling
2. Command loading (including routine table uplink)
3. Science Schedule Execution
4. Science Support Activity Execution
5. Onboard Engineering Support Activities (housekeeping, infrastructure, ground interface, utility support functions, onboard calibration, etc.)
6. Downlinked Data Capture
7. Data and Performance Monitoring
8. Fault Diagnosis
9. Fault Correction
10. Downlinked Data Archiving
11. Engineering Data Analysis/Calibration
12. Science Data Processing/Calibration

Currently, many of the activities listed above are partially automated (for example, planning & scheduling and data and performance monitoring), and may well be fully autonomous (either within the ground or flight systems) in the next 10 years. Some of these functions are already largely performed autonomously onboard (Truszkowski, 1999).

Current State of LOGOS

The current development of LOGOS has established a successful initial prototype system and the efforts have now shifted to a Phase2 development where particular emphasis is begin placed on increasing the functionalit of the individual agents and the agent community in problem solving in a ground system operations context. Figure 1 depicts the current LOGOS architecture that identifies the ten distinct agents. As can be seen from this diagram the multi-agent system is responsible for all operations in the ground system environment. A detailed explanation of the exact operations of the ground system center and each agent can be found in (Truszkowski, 1999).

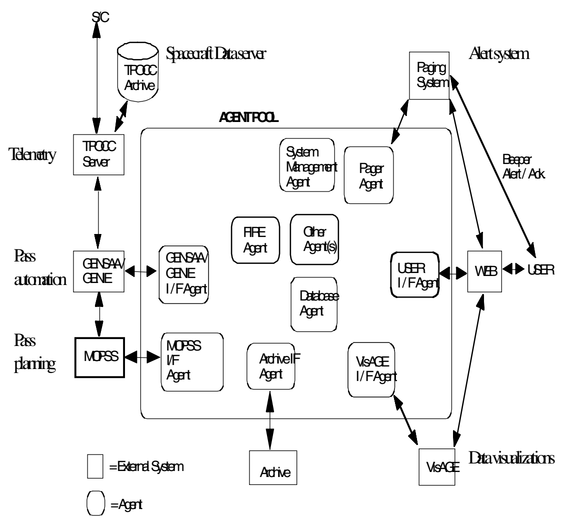

Figure 1: LOGOS (Phase1 Architecture)

One important characterization that can be made about the ten agents within LOGOS is a separation of the agents by their main type of behavior. Table 1 shows the ten named agents and their particular categorization as either reactive or deliberative agents.

Reactive Agents	Deliberative Agents
Archive Interface Agent	FIRE Agent
Database Interface Agent	User Interface Agent
GenSAA Interface Agent	VisAGE Interface Agent
Log Agent	System Management and Monitoring Agent
Pager Agent	
MOPSS Agent	

Table 1: Classification of LOGOS Agents

Each of these agents has a distinct functionalit within the lights-out ground operations context. One method that has been proposed for increasing the agent functionality, particularly the deliberative agents, is to incorporate various learning technologies. The focus of this paper is on the on-going development of the System Management and Monitoring Agent (SysMMA) and the design and use of specific learning technologies within this agent for the long-term success of LOGOS.

Why Learning?

Multi-agent systems are a combination of heterogeneous computing elements that are fashioned together with the intention of being able to cooperate with one another to solve problems that singular, monolithic systems can not solve due to complexity. The advent of agent-based programming and systems development utilizes the distributed, networked nature of many problems or organizations and attempts to mimic group problem-solving behavior. Man advantages have been attributed to multi-agent systems, but the goal is to produce "communities" of problem-solving entities that are adaptable, flexible, "robust", and eventually creative, in solving large complex problems. A "robust" agent is defined as an agent that has the ability to reason in a deliberative and/or social fashion, plan, schedule, model the environment, and learn (Truszkowski, 1999).

In order for these multi-agent systems to be adaptable and robust, the community of agents, as well as the agents themselves, must have to ability to gain knowledge from their experiences in solving problems. Learning can be thought of as the process of translating (or transforming) observations or experience into knowledge to be stored in a form suitable for use whenever needed (Fabris, 1998). To date, many different learning techniques have been developed and implemented; see (Honavar, 1994) for more details.

SysMMA: The Control Agent

The System Management/Monitoring Agent (SysMMA) is responsible for monitoring and managing all agents in LOGOS and serves as the main control agent within LOGOS. In the initial LOGOS prototype, SysMMA fulfilled the following requirements:

1. Cataloged information on each registered agent in the Agent Registry file. The same Registry is also used to provide, upon request, the capabilities of agents that exist within the system as well as the name/location of the agent that is able to perform the requested function.

2. Acted as an agent Health and Safety Monitor. SysMMA kept track of agent status by subscribing to the status update publication in Workplace. Workplace is the name given to the overall system environment that the LOGOS prototype was developed within. This environment provided the basic message passing protocols for the multi-agent system.

3. Had responsibility for sending begin-activit warning messages before the spacecraft pass contact starts. SysMMA did this by querying the MOPSS Interface Agent (MIFA) for the latest schedule information and keeping track of any new

schedule updates. SysMMA also sent an end-of activity message at the end of the spacecraft pass contact.

In fulfilling these requirements, SysMMA acted as the system's Yellow Pages agent and was responsible for monitoring and managing all agents in the system. It received registration messages from LOGOS agents upon the agent's initialization. This message includes the agent's name, its capabilities, initial status and location. SysMMA also indirectly monitored the agents' health and activities by subscribing to the WorkPlace's status publication services, and occasionally polls the agent to determine its latest status. The agents' status and capabilities information is stored in Agent Registry and is used by SysMMA to provide replies to LOGOS agent services requests.

SysMMA was also responsible for some event timing, task completion and resource utilization management. It kept track of the latest spacecraft schedule information and sent warning messages to all agents in LOGOS several minutes prior to the start of a spacecraft "pass", thereby giving agents enough time to finish up any "unimportant" task and get ready for the pass. A spacecraft "pass" is defined as the time interval in which the ground station has the ability to communicate with the spacecraft and visa versa.

SysMMA also broadcasts end of pass message upon the reception of LOSS_OF_SIGNAL to inform the agents to stand down from a spacecraft pass. Finally, SysMMA also kept an up-to-date activit record of each agent. The major deficiency of SysMMA in Phase1 was that the agent lacked the capability to learn.

In the future the role of SysMMA will be expanded. Not only will SysMMA, or a small sub-agent of SysMMA, handle the duties outlined above, but SysMMA must be able to manage (in a near real-time environment):

1. the system utilization of resources available to LOGOS,

2. the completion of tasks by individual agents, and,

3. most importantly, the overall missions completion for the spacecraft pass.

This functionality is summarized by the following future requirements (NASA GSFC, 1999):

1. SysMMA will be able to start/stop other agents in the system,

2. SysMMA will be able to poll an agent's latest status without having to wait for an update message, or, if the agent was not heard from for a certain amount of time,

3. SysMMA will be able shutdown/restart an agent based on system status or when SysMMA ha determined that an agent has stopped responding to ACL messages, and

4. Event level monitoring will be implemented. SysMMA will be able to "recommend" actions to the agents according to the global performance goals of the LOGOS system.

Given this great level of responsibility and importance within LOGOS, SysMMA must be able to recognize trends in task completion and resource utilization to anticipate potential work slowdowns o bottlenecks in performing spacecraft monitoring.

Learning in SysMMA

There are many learning techniques that could be incorporated into SysMMA to aid in its problem solving activities within LOGOS.

These techniques include case-based reasoning, neural networks, statistical pattern recognition, genetic algorithms, and reinforcement learning. An essential element to the development of learning within SysMMA, and other agents within LOGOS, is the application of the "correct" learning components or processes. In addition, each of the heterogeneous learning technologies (in terms of their knowledge representation) must be able to share their increased knowledge with other parts of the agent and the remainder of the community. We do not address this issue in this paper. Next, however, we will examine each learning technology in light of the future requirements of SysMMA.

The need to start/stop individual agents is to allo for: (1) the readying agents prior to their actual usage (this allows some agents to "sleep" during a system pass), (2) the efficient utilization of system resources and (3) to ensure that agents are performing their tasks in a timely manner. Case-based reasoning (CBR techniques can be utilized to aid in this task. For example, SysMMA can recognize a situation (i.e., case) when greater levels of expertise are needed to identif and resolve anomalies. In this case, the FIRE agent could be started and queried for input into the anomal resolution. If, after some processing time has passed, the SysMMA agent could recognize (again using CBR) to start a Pager Agent which is used to summon a human expert to help solve the anomaly not handled b the lights-out system.

Since SysMMA is ultimately responsible for the health and safety of the agent community, all agent must be functioning and performing their assigned tasks within interruption. However, situations occur when various system resources become unavailable and must be restarted in order to ensure the completion of the overall mission. In the current Phase1 development, the SysMMA agent would periodically "ping" each of the agents within the system to ensure that they were still on-line. This was a waste of system resources. In future versions, other methods to ensure the health of the agents must be derived. One such method involves the monitoring of task completion at the agent level and the estimation of the time to completion of various tasks. This type of knowledge could be learned by SysMMA through the use of statistical pattern recognition and neural network techniques. For example, if a particular task an agent had undertake was normally estimated (found through statistical studies) to take X±Y minutes and the agent had not completed the task after Z minutes (where Z is greater than X±Y) than the agent could use a trained neural network to determine if an action against that agent should be undertaken. By training a neural network to understand if the behavior exhibited by the agent was within "normal operational conditions" excess "pinging" of the agent could be avoided.

The ultimate goal of the LOGOS system is to "replace the human operators" that currently monitors the health and safety of the spacecraft. The degree to which LOGOS can perform to the level of expertise of the human beings will be the ultimate standard against which it is judged.

While each agent, by definition, is autonomous (Truszkowski, 1999), the actions of the agents must be coordinated with the rest of the community in order to achieve a high degree of success with regard to th overall mission. SysMMA will have responsibly fo the monitoring the actions of the agents (both present and future planned actions) and must be able to judge which actions are meaningful in regard to fulfillment of the mission. This requires SysMMA to formulate a high level plan and to assign the plan activities (i.e., sub-goals) to the agents most capable of handling those activities. Also, SysMMA must ensure that those activities are being carried out. Each agent will have the ability (if necessary) to formulate plans and to recognize when plans are going to fail, however, it is the responsibility of the SysMMA agent to match the sub-goals with the particular capabilities of the agent in an efficient and effective manner.

The use of case-based planning techniques, neural network technologies, and/or genetic algorithms could be utilized to ensure that the sub-goals are properl distributed among the agents. In addition, a neural network control system could be utilized to process each of the sub-goals upon completion to assess the completion of the overall mission.

Another learning technique that seems most promising (and most technologically challenging) is reinforcement learning. Using reinforcement learning (a subset of supervised learning) each of the individual agents is weighted as to its contribution toward the overall goal of the system (Sun, 1999) (in this particular domain it is the successful completion of mission of the spacecraft). This weighting could be modified over the course of the operational pass to ensure that the overall mission success is ensured. Agents performing their tasks at a higher level (they are weighted higher) could be given more responsibility during the current spacecraft pass.

Future Directions

NASA is committed to the on-going investigation of agent technologies to monitor spacecraft and solve system problems that are currently handled by human beings. This drive toward "faster, better, cheaper" is an environment that NASA will and must live with for the foreseeable future.

Given this drive toward automation the development of LOGOS is vital to the future success of NASA and scientific missions. NASA, and especiall the Goddard Space Flight Center, will continue to develop and enhance the LOGOS prototype and eventually implement a version of LOGOS within an operational context.

One of the key factors in the success of LOGOS is the identification of agents that need specialized learning technologies for their future deployment. In this paper we discussed one such agent within LOGOS, SysMMA, and its need for learning capabilities as currently envisioned. Once the identification of situational learning is complete implementation of these learning technologies can be accomplished. After identifying the various situational learning technologies within each agent, a method must be developed to shar newly learned knowledge within the agent and in the entire LOGOS community. This question is currentl under study with regard to a new single agent architecture that was developed at Goddard over the past year (Truszkowski, 1999).

SysMMA is vital to the success of LOGOS as the main control agent. NASA Goddard will continue to look for further opportunities to expand the capabilities of SysMMA and identify situations in which learning can and should be utilized. After deployment of the prototype further study will need to be undertaken to ensure that a final agent-based system meets the needs of technical and scientific personnel in spacecraft future missions.

Acknowledgements

The author would like to thank the personnel in the Advanced Automation and Architectures Branch at the NASA Goddard Flight Center for their on-going support of these research efforts. In addition, this dedicated team of professionals has provided many insights and new avenues for study by monitoring the progress of this work and asking articulate and foundational questions with regard to system implementation issues.

References

Fabris, P. Data Mining. *CIO Magazine*. May 15, 1998.

Honavar, V. 1994. Toward Learning Systems That Integrate Different Strategies and Representations. In *Symbol Processors and Connectionist Networks for Artificial Intelligence and Cognitive Modelling: Steps toward Principled Integration.* Honavar, V. and Uhr, L. eds. New York: Academic Press.

NASA Goddard Space Flight Center. 1999. SysMMA Design Document. http://agents.gsfc.nasa.gov/documents/lockheed-martin/sysmmadesign.rtf.

Sun, R. and Peterson, T. 1999. Multi-agent reinforcement learning: weighting and partitioning. *Neural Networks*.

Truszkowski, W.; Hallock, H.; and Remote Agent Team (NASA Ames), 1999. Agent Technology from a NASA Perspective. Invited Talk. In Proceedings of the Third International Workshop of Cooperative Information Agents. Menlo Park, Calif.: AAAI Press.