Engineering Multigroup Agents for Complex Cooperative Systems

(Doctoral Consortium)

Denise M. Case
Multiagent and Cooperative Reasoning (MACR) Laboratory
Kansas State University, College of Engineering
Manhattan, KS USA 66506
dmcase@ksu.edu

ABSTRACT
Intelligent agents and multiagent systems (MAS) provide a scalable approach to distributed artificial intelligence [3]. Analogous to intelligent biological entities, agent-based computational systems may become quite complex. Some may consist of multiple cooperating groups, each providing a specialized subset of the total required functionality [4, 6]. My work focuses on engineering multigroup agents in cooperative intelligent systems. A reusable multigroup agent architecture has been developed, along with a simulation framework and recommended software engineering practices, to support the implementation of multigroup applications. The architecture is being used to evaluate power quality control algorithms for electrical power distribution systems (PDS).

Categories and Subject Descriptors
D.2.11 [Software Engineering]: Software Architectures

General Terms
Design

Keywords
Agent architectures, organization-based agents, distributed artificial intelligence, agent-oriented software engineering

1. INTRODUCTION
Agents are autonomous interactive software entities that may work cooperatively or competitively within the structure of a MAS [4]. The behavior of the MAS is typically driven by a set of goals that are generally decomposed and ultimately assigned to individual agents [8].

In some MAS applications, it may be helpful to design the system as a collection of specialized groups [5], a system we refer to as a complex MAS. In a complex MAS, each group may have its own set of specialized goals, and the groups work cooperatively to achieve overall objectives.

Multigroup agents participate in multiple groups and may get assignments from more than one. Standard mechanisms can be helpful for assessing goal consistency and managing the appropriate evaluation, negotiation, and acceptance of tasks before incorporating them into the agent’s workflow.

A multigroup agent may represent a complete group within a different group. In some cases the groups may be designed recursively, and agents may form remote groups of more distributed agents with an agent representing the distributed organization in a higher-level, more centrally-located group. In our approach, we have designed our multigroup agents as both an organization and an agent. This concept of an agent as an organization provides the foundation for our approach to managing the complexity of multigroup agents through implementing a reusable agent architecture.

2. AGENT ARCHITECTURE
Our architecture builds on the Organization-based Agent Architecture (OBAA) [9] designed to support reactive and proactive organization-based agents. Each OBAA agent has a general-purpose Control Component (CC) and an application-specific Execution Component (EC), allowing separation of the general features needed for group participation from the execution of application objectives.

The goal of the multigroup OBAA architecture is to extend similar reusability to multigroup agents. To do so, we’ve designed each OBAA++ agent as a goal-driven self organization of sub-agents [2]. To clarify the distinction between the agent and its sub-agents, we introduce the term persona for the sub-agents. Each agent has one persona for each affiliated group and one self persona that acts as the central, goal-directed brain of the agent, responsible for initiating and managing the roles and commitments within that affiliated group. Each agent is initialized with a set of goals that drives the behavior of the self persona. They include


Figure 1: OBAA++ agent architecture
goals for joining and maintaining membership in specified affiliated groups. When a self persona determines that its agent requires a new affiliation, the self persona creates a new persona to join the group. Self persona function similarly for each agent in a complex MAS.

OBAA++ assists the designer by providing some key communication and control features as reusable components, sub-agents, and capabilities. Agents can be designed, implemented, and tested individually by supplying appropriate configuration information including the list of affiliated groups, whether the agent is responsible for initially running any of those groups, and how the agent should contact and begin participating in the group. Application-specific behavior is designed and implemented as in OBAA using goals, role behaviors, and capabilities.

3. AOSE & MAS SIMULATION

To enable efficient development and feedback on increasingly complex test cases, my work includes the development of the AO-MaSE Agent-Oriented Software Engineering (AOSE) process, an agile-based adaptive methodology compliant with the Organization-based Multi-agent System Engineering process [1] that provides architects and developers a structured approach for iteratively adding functionality to create complex adaptive systems.

4. POWER DISTRIBUTION HMAS

OBAA++, AO-MaSE, and the MAS simulation framework are being used as the basis for the Intelligent Power Distribution System (IPDS) project that aims to evaluate control algorithms in electrical power distribution systems. The work involves the development of a holonic multiagent system (HMAS). Holonic is the merger of holos (whole) and on (parts) [7], and indicates a single agent may recursively represent an entire lower group of agents while acting as part of a higher-level group. This allows our agents to optimize locally in lower-level distributed groups first while progressively optimizing towards increasingly higher-level centrally-viable solutions. A portion of the IPDS HMAS simulation is shown in Figure 2; current test cases include 62 hosts, 124 agents, 40 organizations, and 110 inter-agent connections.

5. INTERDISCIPLINARY ASPECTS

My research in cooperative intelligent systems supports several interdisciplinary interests. Our architecture was designed to emulate installation in intelligent cyber-physical systems (CPS). Communication, cooperation, validation, and management of goal consistency are crucial. Inspiration comes from recent work in the areas of complex biological systems, human and agent cooperation, game theory, machine learning, and natural and artificial intelligence.

6. FUTURE WORK

Future work is focused on managing goal consistency among multigroup agents supporting the objectives of multiple groups while reflecting differing degrees of selfishness and cooperation. We hope to develop additional architectural aspects and processing algorithms to assist with managing goal consistency in complex systems. We hope to extend the functionality of the evaluated control algorithms in the IPDS project and are planning on scaling test cases to 400 hosts and 800 agents to evaluate the effects of spatial correlations in the distributed control application and further assess the scalability of the IPDS architecture. We also plan on extending the agent-based simulation framework to include additional MatLab-powered sensor simulators and actuator adapters as a configurable set of standard interfaces to the simulated environment.

7. ACKNOWLEDGMENTS

We thank the US National Science Foundation for providing support for the work via Award No. CNS-1136040.

8. REFERENCES