OBAA++: An Agent Architecture for Participating in Multiple Groups

(Extended Abstract)

Denise M. Case
Kansas State University
234 Nichols Hall
Manhattan, KS USA 66506
dmcase@ksu.edu

Scott A. DeLoach
Kansas State University
234 Nichols Hall
Manhattan, KS USA 66506
sdeloach@ksu.edu

ABSTRACT

Organization-based agents dynamically create, manage, and participate in groups governed by common goals. In complex applications, agents may participate simultaneously in multiple organizations, requiring agents to manage their associations and assignments in a consistent, unambiguous way. The Organization-Based Agent Architecture ($OBAA^{++}$) explicitly supports *multigroup agents* and is being used to evaluate a holonic MAS for power quality control.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Design

Keywords

Agent architectures, organization-based agents, self-adaptive systems, agent-oriented software engineering

1. INTRODUCTION

For some multiagent (MAS) applications, a *complex MAS*, where the MAS is decomposed into multiple groups may be helpful [5]. In some of these complex MAS applications, agents may need to participate in more than one group simultaneously [4]. These *multigroup* agents must be able to accept and successfully manage goal and task assignments from different affiliated groups. Agent organization is an active area of research with ongoing efforts to develop and evaluate state-of-the-art models, frameworks, and infrastructures to support organization-based MAS. Hubner et al. have developed an organizational middleware (S-Moise+) providing support for both heterogeneous and open MAS [6]. Dignum introduced a framework for organizational interaction, OperA, that focuses on societal communications, coordination, and common goals [3]. The JaCaMo model and infrastructure provides support for belief-desire-intention (BDI) Jason agents operating in Moise organizations [2] and Ribino, et al. have been developing norm-governed MAS meta-models [7] for industrial holonic applications. Engineering systems capable of handling complex, real-world structures is challenging and flexible approaches are required [8].

The goal of the $OBAA^{++}$ architecture is to support multigroup agents. This was accomplished by designing each $OBAA^{++}$ agent as an organization of sub-agents. To clarify the distinction between an agent and its sub-agents, we introduce the term *persona* to describe the sub-agents.

The architecture is being used to evaluate power distribution system (PDS) control algorithms. PDS are well-suited to MAS. PDS carry electricity from generation facilities to customers through distribution lines and transformers and may be hierarchically and radially distributed, with cross-ties and interconnects, requiring complex interaction patterns. Research in intelligent PDS provides a variety of opportunities for complex MAS [1, 9].

2. OBAA++ AGENT ARCHITECTURE

The key element of the $OBAA^{++}$ architecture is the internal organization of personae. Each agent has exactly one persona for each affiliated group. Each persona is implemented with the OBAA architecture and includes a CC-EC pair as shown in Figure 1. This agent has two affiliated groups: n and $n-1$. The Level n persona participates in the body of the Level n group while the Level $n-1$ persona acts as head of the Level $n-1$ group. The *self persona* provides the internal management of the agent and is not affiliated with an outside group.

Designing and building multigroup agents can be more difficult than developing traditional agents because a multigroup agent must be able to align its goals and behavior with the goals assigned by various affiliated groups. To support

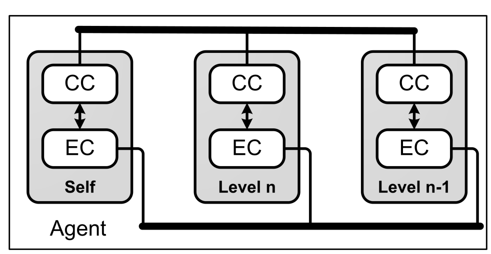

Figure 1: $OBAA^{++}$ **agent architecture**

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

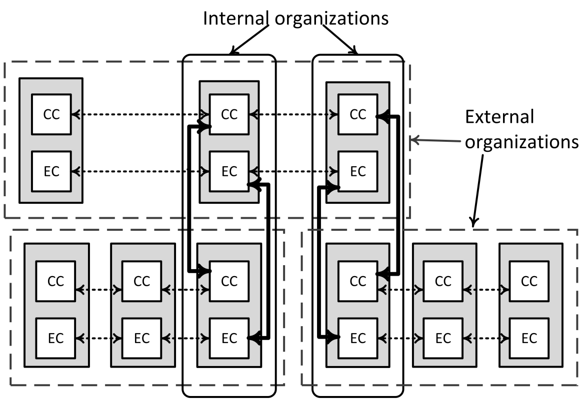

Figure 2: Communications in $OBAA^{++}$

this alignment of goals and behaviors, each $OBAA^{++}$ agent operates as an organization of persona, where each persona is focused on establishing membership and performing the appropriate behavior in a single affiliated group.

The internal organization of each agent includes a unique *self persona* that acts as the head of the agent's personal organization of personae. During instantiation, the self persona is created and given a set of goals to drive the behavior of the self persona. Provided goals are application-specific and include goals for discovering, joining, and maintaining membership in the desired groups. For each group participation goal, the self persona will instantiate a new persona to join that group. The self persona is designed to provide this functionality in a standard manner and functions identically for each agent in a complex MAS.

$OBAA^{++}$ allows an agent to participate in multiple affiliated groups without burdening the designer with details of the interactions of the various groups. Each persona can be designed, implemented, and tested individually and incrementally by supplying appropriate agent configuration information specifying the groups to join, whether the agent is responsible for creating and administering each group, and authorization and authentication information for contacting and participating in the group.

$OBAA^{++}$ agents communicate along two dimensions: externally and internally. As shown in Figure 2, external organizations may be considered to be oriented horizontally (encompassed by dashed rectangles) with internal agent organizations oriented vertically (encompassed by solid rectangles). For clarity, self personae have been omitted.

As shown, each persona can communicate internally with other personae and externally with other agents in the affiliated group. *Within an agent*, persona communicate by passing messages via internal communication queues, shown as bold arrows vertically within the internal organization. The persona Control Components (CCs) communicate via the CC communication queue, while persona Execution Components (ECs) communicate via the EC communication queue, supporting standard, administrative group communications and application-specific task execution, respectively. OBAA communication between the EC and CC within the persona allows events and assignment information to be transmitted within the CC-EC pair. Communication *between agents* is shown as dashed arrows between the CCs and ECs of persona in the external organizations. As with the internal communications, CCs exchange administrative organizational knowledge and ECs exchange application-specific information during the execution of assigned tasks.

3. APPLICATION

$OBAA^{++}$ is being used as the basis for the Intelligent Power Distribution System (IPDS) project that aims to evaluate distributed control algorithms for PDS. The system is designed as a holonic MAS, providing distributed local solutions that can be propagated up the distribution hierarchy. The JVM-based system includes integrated MATLAB power flow calculations. The goal is to make the cyber-architecture realistic so it could be used in a cyber-physical deployment with distributed computation.

The test configuration includes 62 hosts, 124 agents, 46 organizations, and 110 inter-agent connections. Plans exist to expand the tests to include cases with up to 400 hosts and 800 agents to assess the effects of spatial correlations in the distributed control application and to further evaluate the scalability of the IPDS architecture.

4. CONCLUSIONS

This paper presents the $OBAA^{++}$ agent architecture for implementing multigroup agents in complex MAS. As far as we know, $OBAA^{++}$ is the first agent architecture specifically designed to support multigroup agents.

The work is supported by the US National Science Foundation via Award No. CNS-1136040.

5. REFERENCES

[1] G. Basso, V. Hilaire, F. Lauri, R. Roche, and M. Cossentino. A mas-based simulator for the prototyping of smart grids.

[2] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 2011.

[3] V. Dignum. *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, University of Utrecht, Utrecht, The Netherlands, 2003.

[4] O. Gutknecht and J. Ferber. Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems. *Rapport Interne LIRMM*, 97188:1997, 1997.

[5] B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2004.

[6] J. F. Hubner, J. S. Sichman, and O. Boissier. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *Intl Journal of Agent-Oriented Software Engineering*, 1(3):370–395, 2007.

[7] P. Ribino, C. Lodato, S. Lopes, V. Seidita, V. Hilaire, and M. Cossentino. A norm-governed holonic multi-agent system metamodel. 2013.

[8] S. Rodriguez, V. Hilaire, and A. Koukam. Formal specification of holonic multi-agent systems framework. In *Computational Science–ICCS 2005*, pages 719–726. Springer, 2005.

[9] H. Wang. Multi-agent co-ordination for the secondary voltage control in power system contingencies. In *In Proc. IEE Generation, Transmission and Distribution*, pages 61–66. IEEE, Jan 2001.