From: AAAI-00 Proceedings. Copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Tracing Dependencies of Strategy Selections in Agent Design

Dung N. Lam, K. S. Barber

The Laboratory for Intelligent Processes and Systems
Department of Electrical and Computer Engineering, ENS 240
University of Texas at Austin, Austin, TX 78712-1084
{dnlam, barber}@lips.utexas.edu

Given the diverse multi-agent system (MAS) implementations developed for various domains, there has been a lack of a comprehensive method for analyzing and evaluating the assortment of MAS architectures and technologies resident in those architectures. With a formal method to investigate agent architectures, MAS designers can answer 1) how performance criteria affect design decisions, 2) how design decisions affect MAS behavior, and 3) which combinations of design decisions are best suited for the application. This research proposes that the first step in answering these questions is to decompose an agent into its *core competencies* (CC), which define the major functionalities of an agent. Some example core competencies include agent organization (AO), plan generation (PG), task allocation (TA), plan integration (PI), plan execution (PE), world modeling, communication, actuation, and perception (Barber, Liu, and Han 1999). CCs affect the behavior of the agent and of the system.

For each CC, the designer chooses to implement a *core competency strategy* from a library of existing strategies. With each design decision, the number of possible strategy combinations decreases due to the dependencies among strategies across CCs. In solving a given problem, an agent coordinates itself with other agents in the system to organize and to create a plan and/or reactively generate actions that leads to a solution. The agent uses its chosen agent organization, which defines how agents interact with each other, to manage the agents during the planning process. First, the agent works alone or cooperates with other agents in PG. Next, in TA, those plans and subtasks are distributed to the appropriate agents. Finally, in PI, the agents' schedules are combined according to the chosen strategy, which results in task-coordinated agents. After planning, the agents must monitor the execution of the subtasks and make any necessary adjustments.

Each CC strategy has dependencies based on its *demands* on the functionality of other CCs (e.g., a market AO demands that TA be composed of proposals, bids, and agreements). Additionally, the selection of a CC strategy is dictated by dependencies among *variables* (factors and properties that affect the choice of strategy used), such as available resources and the number and type of agents involved. CCs are instantiated as CC strategies and are illustrated in Figure 1 as a sequence of adaptors that links the problem to a solution. Each CC strategy adaptor is built upon the previous adaptor and thus is constrained by the dependencies of the previous CC strategy. The male pin connectors represent the variables and demands of that CC strategy that can be passed on to the next CC strategy. The female pin connectors suggest the demands that are passed on from the previous CC strategy. The pin connectors must match, or the demands must be satisfied, for CC strategies to be compatible. It is possible to have an adaptor that encompasses more than a single CC, such as Partial Global Planning.

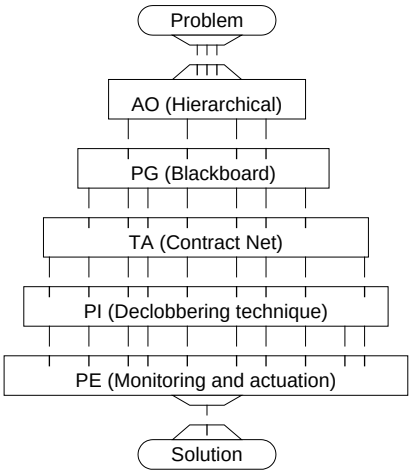

Figure 1: CC strategies are shown as adaptors that build upon previous adaptors.

With the resulting method to analyze agents at an abstract level, MAS designers can investigate the system-level implications of selecting strategies that have dependencies spanning across multiple CCs and across agents. Design decisions can be traced from the domain performance criteria that motivated the decisions to the resulting MAS behavior. Continuing research will develop a fundamental understanding of how and why certain combinations of strategies produce specific agent-level and system-level behaviors.

References

Barber, K. S., Liu, T. H., and Han, D. C. 1999. Agent-Oriented Design. Multi-Agent System Engineering. In *Proceedings of the Ninth European Workshop on MAAMAW*, 28-40. Berlin, Germany: Springer.

Copyright © 2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.
This research was supported in part by the Texas Higher Education Board (#3658-0188-1999).