Achieving Agent Coordination via Distributed Preferences

Joseph G. D'Ambrosio & William P. Birmingham

The University of Michigan
Artificial Intelligence Lab, EECS Department,
Ann Arbor, MI 48109
jdambros@eecs.umich.edu

Agent-based systems provide hope for solving a wide variety of distributed problems. One key aspect of agent-based system is coordinating agent actions to achieve coherent behavior. For example, in concurrent engineering (CE), it is necessary to ensure that the individual decision made by constituents in a design organization achieve overall organizational objectives (e.g., increase market share), while still allowing individuals to exploit their expertise. We believe CE is representative of many multi-agent problems, in that agent coordination must include facilities to support both solving a hierarchically decomposed problem, e.g., the contract net (Davis & Smith 1983), and interactions among peers (Bahler et al. 1995) as well.

One method to support problem decomposition is through task subcontracting (Davis & Smith, 1983). The process of subcontracting creates a hierarchical, decision-making organization. Coordination is accomplished by allowing general contractors, who have a global perspective, to provide direction to subcontractors, who have local expertise.

We believe that it is both practical and desirable to direct a subcontractor's actions by explicitly stating a preference structure that the agent must follow. Just as any contract describes a set of constraints that must be met for the contract to be fulfilled, a contract should include a preference structure that must be followed by a subcontractor in order to fulfill its obligations. Specifying a preference structure for a subcontractor does not require the general contractor to reveal all of its preferential knowledge to the subcontractor, only the subset of preferences that rank the implementation domain of the contracted subtask need be specified.

We are developing a concurrent-engineering tool, ACME (D'Ambrosio, Darr, & Birmingham 1996), which provides a framework for coordinating design agents based on contracting constraints and preferences among agents. Agent preferences are formally represented using a form of a utility function referred to as an imprecise multi-attribute value function. This function has the desirable property that the amount of work required to construct the function is significantly less than that normally associated with defining a multi-attribute utility function.

Consider the following example related to the design of a powertrain control system for an automobile. The project leader establishes contracts with several agents, including an electrical-system agent, and a software-development agent. The project leader's preferences are represented by a value function based on the component costs of the powertrain controller and the expected fuel economy. In general, these preferences provide only a partial order on a subordinate's decisions, and the subordinate is free to exploit its own expertise (preferences) in choosing among alternatives for which the supervisor is indifferent to.

In this example, the electrical system agent is responsible for selecting the necessary electrical components. This agent is primarily concerned with minimizing component costs, and the agent specifies an appropriate value function to represents its preferences. Since the contract between the project leader and the electrical agent includes the project leader's preferences, the electrical agent's decision making is not based on its value function alone, but on a lexicographic value function, where the first and most significant attribute is the value assigned by the supervisor's value function. Given that the supervisor's value function is based on fuel economy as well as cost, the electrical agent decisions will be made in a manner consistent with global preferences, as will all of the decisions made by the other agents in the design team.

The second attribute in the electrical agent's lexicographic value function is the value assigned by a group value function, that contains an attribute for the value assigned by each member of the design team, e.g., the electrical agent and the software agent. The group value function creates the necessary relationship among peers to ensure that the value of composite decisions is maximized in terms of the peer's stated preferences.

References

Bahler, D., et al. 1995. Mixed quantitative/qualitative method for evaluating compromise solutions to conflicts in collaborative design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 9:325-336.

D'Ambrosio, J.; Darr, T.; and Birmingham, W. 1996. Hierarchical Concurrent Engineering in a Multiagent Framework, *Concurrent Engineering: Research and Applications*, Forthcoming.

Davis, D., and Smith, R. 1983. Negotiation as a Metaphor for Distributed Problem Solving, *Artificial Intelligence* 20: 63-109.