Chapter XXII
Organizational Self–Design in Worth–Oriented Domains

Sachin Kamboj
University of Delaware, USA
Keith S. Decker
University of Delaware, USA

ABSTRACT

This chapter presents an approach to organizational-self design (OSD), a method of designing organizations at run-time in which the agents are responsible for generating their own organizational structures. OSD is especially suitable for environments that are dynamic, albeit slowly changing. Such environments preclude the use of static, design-time generated, organizational structures, and yet contain certain characteristics and conditions that change slowly, if at all, and these characteristics can be harnessed for the purposes of creating stable organizational structures. This chapter extends the existing approaches to OSD by applying them to worth-oriented domains – that is, domains in which problems are represented using TÆMS based task structures. This chapter presents our OSD primitives and framework and discusses some interesting future lines of research.

INTRODUCTION

Multiagent systems are increasingly being used to solve a wide variety of problems in a range of applications such as distributed sensing, information retrieval, workflow and business process management, air traffic control and spacecraft control, amongst others. Each of these systems has to be designed at two levels: the micro-architecture level, which involves the design of the individual agents, and the
Organizational Self-Design in Worth-Oriented Domains

At the organizational level, the multiagent designer is primarily concerned with issues such as the number of agents needed to solve the problem, the task structure (i.e. the breakup of the problem into subgoals), the task and resource assignments to the individual agents and the coordination mechanisms to be used. These issues can be resolved by choosing an organizational structure and by instantiating that structure with actual agents. The organizational structure consists of roles that the agents play and the manner in which they interact with other agents in the system. The instantiation consists of selecting the number of agents needed in the system and the assignment of roles and resources to the individual agents.

The organizational structure employed directly influences the effectiveness of the organization in solving the problem at hand, the resources needed by the agents and the cost of coordinating the activities of the individual agents. Hence, the organizational design is a very important part of the multiagent system design. However, there are few good rules and formal mechanisms for designing effective organizations for computational agents that are general enough for a wide range of agent systems. For example, consider the question of the number of agents needed in the system. If too few agents are available, the system will be overloaded and will not be able to perform optimally. If too many agents are used, resources may be wasted and contention for the limited resources amongst the agents will increase.

The macro-architectural design is further complicated by the fact that there is no best way to organize and all ways of organizing are not equally effective (Carley and Gasser, 1999). Instead, the optimal organizational structure depends both on the problem at hand and the environmental conditions under which the problem needs to be solved. In some cases, the environmental conditions may not be known a priori, at design time, in which case the multiagent designer does not know how to come up with the suitable organizational structure. In other cases, the environmental conditions may change requiring a redesign of the agents’ macro-architecture. Hence, it is not obvious that a static design-time approach to an organizational structure is feasible in a significant number of cases. At the opposite end of the spectrum, systems may be designed to create a new, bespoke organizational structure for every problem instance. The most popular example of such a one-off task allocation approach is the Contract Net protocol (Smith and Davis, 1978). Such an approach brings with it a different set of inefficiencies and belies the fact that while many real environments have dynamic components, there are also commonalities in the structure of problem instances that can be taken advantage of through proper organizational structuring.

Organizational Self-Design (OSD) (Corkill and Lesser, 1983; Ishida et al., 1992) has been proposed as an approach to designing organizations at run-time in which the agents are responsible for generating their own organizational structures. We believe that OSD is especially suited to the above scenario in which the environment is semi-dynamic as the agents can adapt to changes in the task structures and environmental conditions, while still being able to generate relatively stable organizational structures that exploit the common characteristics across problem instances.

In our approach, problem solving requests arrive at the organization continuously at varying rates and with varying deadlines. To gain utility, the agents in the organization need to solve the problems by their given deadlines. We start off with an initial organization consisting of a single agent that is solely responsible for all the problem solving activities. As new problem solving requests arrive, the agent checks to see whether it can complete the request by the given deadline. If not, the agent spawns off a
Related Content

Adaptive Collaboration Based on the E-CARGO Model
[www.igi-global.com/article/adaptive-collaboration-based-cargo-model/66069?camid=4v1a](www.igi-global.com/article/adaptive-collaboration-based-cargo-model/66069?camid=4v1a)

Learning Agents for Collaborative Driving
[www.igi-global.com/chapter/learning-agents-collaborative-driving/26941?camid=4v1a](www.igi-global.com/chapter/learning-agents-collaborative-driving/26941?camid=4v1a)

Agent-Based Simulation and Modeling for Service Allocation in Digital Manufacturing Market

Creating a Comprehensive Agent-Oriented Methodology: Using Method Engineering and the OPEN Metamodel
[www.igi-global.com/chapter/creating-comprehensive-agent-oriented-methodology/5065?camid=4v1a](www.igi-global.com/chapter/creating-comprehensive-agent-oriented-methodology/5065?camid=4v1a)