Chapter VII
Describing Agent Societies: A Declarative Semantics

Maksim Tsvetovat
George Mason University, USA

ABSTRACT

Agent-based approaches provide an invaluable tool for building decentralized, distributed architectures and tying together sets of disparate software tools and architectures. However, while the agents themselves have been gaining complexity, and agent specification languages have been gaining expressive power, little thought has been given to the complexity of agent societies, and languages for describing such societies. In this chapter, I propose a declarative language designed specifically for describing in an expressive way a variety of social interactions. I attempt to avoid the fallacies of artificial restriction, and similarly confounding under-specification of the design domain, yet constructing a rigorous, machine-interpretable semantics. It is my hope that introduction of such semantic will lead to a constructive dialogue between communities of agent-based social modeling and agent-based software design, and lead to a greater integration of agent development toolkits and agent-based modeling toolkits.

MOTIVATION

Agent-based approaches provide an invaluable tool for building decentralized, distributed architectures and tying together sets of disparate software tools and architectures. However, while the agents themselves have been gaining complexity, and agent specification languages have been gaining expressive power, little thought has been given to the complexity of agent societies, and languages for describing such societies.

There are two fundamental approaches in design of multi-agent systems. In one approach, collaborative agents of heterogeneous functionalities coexist in a world where cooperation is expected, communica-
Describing Agent Societies

tion protocols are well-defined and interactions are largely scripted by the developers. In the second approach, large numbers of agents perform similar functions (e.g. price negotiations), abide by a small number of standard negotiation protocols (e.g. auctions) and interactions are strategic.

Both of these approaches are characteristic for engineered agent-based systems. By engineered systems I mean systems that are designed, as a whole, to perform a particular task or set of tasks, or to provide an infrastructure for human users of the system to perform their tasks. In engineered systems, efficiency and performance of the tasks are paramount, thus leading to simpler, well-defined interaction patterns. These patterns are generally a result of a distinct design effort, and represent an idealized view of how such interactions might happen. Within this scope of requirements, standard interaction protocols (e.g. FIPA(FIPA, 1997)), and agent communication languages of artificially limited expressive value (e.g. KQML(Finin, et al. 1994)) are appropriate and optimal for design and specification of the agent-based system.

Meanwhile, the social simulation and social network analysis communities have been facing a very different problem. In social simulation, the goal of the designers is to create a facsimile of a real-world social system (e.g. a terrorist network) as a means to study social phenomena in an in-silico experimental laboratory. In this pursuit, the most important notion is that of face validity, i.e. recognizability of interaction patterns and social structures within the context of the experimental subject. Thus, the notions of efficiency of agent interactions and their ability to complete tasks take a back seat to the notions of realistic representation.

Within the community of social modelers, different approaches to modeling interaction complexity exist. Some succumb to simplifying assumptions of grid-based “worlds”, or interaction fields, borrowed from Artificial Life. Others model agent interaction as series of method calls within an object-oriented memory space. Finally, a small group of modelers has adopted the standards of engineered agent systems, and is engaging in design of complex artificial societies around tools that were designed for significantly less complexity of interactions.

Needless to say, all three approaches present their limitations. Grid-world interactions are spatially explicit and thus appropriate for models that take into account geographical features – but at the same time limit numbers and types of interactions possible within a neighborhood. Object-method calls are less limiting in terms of interaction types, but require separately developed design documentation (e.g. UML diagrams) to understand the social system without delving into the source code. KQML and FIPA-based tools present modelers with an opportunity to abstract away the physical interactions of agents and concentrate on modeling the underlying social system - and yet are too limited to allow for construction of highly complex relationships such as friendship, kinship, subordination and discipleship.

In this paper, I propose a declarative language designed specifically for describing in an expressive way a variety of social interactions. I attempt to avoid the fallacies of artificial restriction, and similarly confounding under-specification of the design domain, yet constructing a rigorous, machine-interpretable semantics.

It is my hope that introduction of such semantic will lead to a constructive dialogue between communities of agent-based social modeling and agent-based software design, and lead to a greater integration of agent development toolkits and agent-based modeling toolkits.