Preface

During the last several years, the field of agent and multi-agent systems has experienced tremendous growth, i.e., the topic has gained unprecedented popularity. Meanwhile, the field of formal methods has also blossomed and has proven its importance in substantial industrial and government applications. Thus, in 2000 it was quite timely to pursue a workshop to merge the concerns of the two fields. The need for such a workshop was particularly compelling given the growing concerns of agent-based systems users that their systems should be accompanied by behavioral assurances. The Formal Approaches to Agent-Based Systems (FAABS'00) workshop was the first step in trying to address this need. The overwhelming response to FAABS'00 motivated subsequent FAABS ('02 and '04) workshops, as well as this book, which is designed to provide a more in-depth treatment of the topic.

This book is organized into four parts. Part I provides introductory background material on the two central topics of the book, namely, agents and formal methods.

Chapter 1, by Truszkowski, is an overview of agents. The chapter begins by introducing the basic concept of an agent from a very simple, abstract perspective. It then gradually refines this notion into a detailed agent architecture, using the Goddard agent architecture as an example model. First, the major architectural components (e.g., percepts, effectors, communications, reasoning, planning, execution) are defined and described. Then, agent behaviors are defined and related to the architectural components that generate them. The chapter concludes with an intriguing discussion of multi-agent communities.

Chapter 2, by Hinchey et al., provides an introduction to formal methods, and, in particular, formal methods for agent-based systems. A definition of the term "formal methods" is presented along with overview of the field of formal methods, as well as examples of several formal methods and how they have been used. It begins by describing different classes of formal specification languages and provides an overview of modelling systems and tools. Next, technology transfer and research issues are discussed, describing some of the hurdles that have prevented formal methods from being more fully utilized in industry. The chapter then discusses formal methods that have been used on, or are good candidates for use on, agent-based systems. The chapter concludes with sources of information on formal methods, a list of important terms that are used in the formal methods community, and a wide range of references to related formal methods work.

Chapter 3, by Luck and d'Inverno, gives an introduction to formally specifying agent-based systems. Specifically, it provides a conceptual framework for modeling, analyzing, implementing, and deploying multi-agent systems. The formal specification language used for modeling, Z, facilitates the transition to implementation. The power of this framework is demonstrated in the context of a paradigm consisting of agents that are autonomous (i.e., generate their own goals and motivations), and in which sociological behavior occurs as a result of enabling agents to discover, evaluate, and revise theories about their environment and the other agents within it.

Part II, on formal methods in agent design, focuses on using formal methods to facilitate the task of transitioning from the conceptualization of an agent-based system to its implementation. The organization of this part progresses from more conventional to less conventional techniques.

Chapter 4, by Esterline et al., describes the PI-calculus process algebra for modeling multi-agent systems in which the emphasis is on communication. In this framework, agents are considered to be processes, and the modeling of communication protocols is straightforward. The effectiveness of such an approach is demonstrated by modeling the NASA LOGOS multi-agent system. LOGOS is a prototype automated grounds operation center for flight control, which is described in greater detail in Chapter 10 of this book. The latter portion of Chapter 4 describes the applicability of process algebraic models to multi-agent systems in which one or more agents are human. The agent architecture is assumed to be BDI, and the viewpoint adopted is that of the intentional stance. Again, the emphasis is on communication—in this case, mixed-initiative.

In Chapter 5, Fisher et al. present METATEM, one of the few high-level agent-based programming languages. METATEM is an executable formal specification, based on linear temporal and modal logic. Furthermore, it allows resource-bounded reasoning about belief contexts (e.g., agent A believes that agent B believes that z will eventually occur). Using broadcast message communication, concurrent METATEM can be used for asynchronous executing agents to dynamically form a team to achieve a joint goal.

Part II concludes with Chapter 6. In this chapter, Whittle and Schumann present a method for using "scenarios" as a basis for (semi)automatically synthesizing and explaining agent-based systems. Scenarios are execution traces of interactions between a system's components or its users. The chapter's primary focus is on forward engineering, i.e., using the scenarios for system design. Agent-based systems are represented as statecharts, which are a generalization of finite-state machines that includes models of hierarchy and concurrency. A secondary focus of this chapter is on reverse engineering of agent communication protocols. This novel application of the synthesis algorithm enables (semi)automatic construction of an explanatory model from cryptic code via traces of the code's execution.

Part III is about formal agent verification and re-design. As in Part II, the organization of this section progresses from more conventional to less conventional techniques. The verification techniques include theorem proving, model checking, and average case analyses.

Part III begins with Chapter 7, by Hustadt et al. Research on practical proof methods for expressive theories of rational agency has been sparse. This chapter helps to fill this gap by focusing on how to apply formal verification, in particular, theorem proving, within the exceptionally expressive KARO agent framework. Two proof methods are presented in this chapter. The first consists of translating KARO modal formulae to first-order logic and then using conventional first-order theorem proving techniques. This translation morphism has the advantage of being an elegant treatment of the informational component of KARO. The second method first translates the dynamic component of KARO into a branching time temporal logic, but leaves the epistemic component unchanged. Then, clausal resolution-based theorem proving is applied to the branching time temporal logic. An advantage of this latter method is its potential to provide a complete calculus for the dynamic component of KARO.

Chapter 8, by Spears, addresses the issue of formally verifying multiagent systems in which the agents can adapt. Agent plans/strategies are represented with finite-state automata, and they are adapted using evolutionary learning. Verification is performed using model checking. Two approaches are presented for behavioral assurance following adaptation. The first consists of a priori results that certain learning operators are guaranteed to be property-preserving. For these operators, no reverification is needed. Second, incremental reverification algorithms are presented, along with time complexity results demonstrating that they can produce substantial speedups over nonincremental reverification (i.e., reapplication of traditional model checking).

In the concluding chapter of Part III, Chapter 9, Menzies and Hu present a nonstandard formal analysis that makes a nondeterminism assumption. Given a group of agents working in a highly dynamic environment, it is important to verify whether their behavior is correct, despite "wild" influences (e.g., random perturbations). Menzies and Hu address this issue by defending two claims. The first claim is that in the average case, it is possible to learn effective and efficient control strategies despite the wild variables. Using an average case analysis involving the combinatorics of possible worlds of belief, Menzies and Hu present a case that supports the first claim, e.g., they show that in most circumstances only a small number of different behaviors are possible, despite the wild inputs. Their second claim is that it is possible to redesign particular devices in order to increase their immunity to wild variables. This second claim is supported by a sensitivity analysis.

Part IV, on significant applications, presents two substantial and exciting NASA applications. The chapters in this section demonstrate the power of using formal methods for agent-based systems for solving important, real-world problems.

Chapter 10, by Rouff et al., describes a successful NASA Goddard Space Flight Center (GSFC) application of formal methods to a multi-agent system called "Lights-Out Ground Operations Systems" (LOGOS). In LOGOS, a community of software agents cooperates to perform satellite ground system operations normally done by human operators. Due to the asynchronous, parallel nature of both the agents and their communications, debugging LOGOS using traditional techniques is difficult or impossible. On the other hand, by specifying LOGOS within the Communicating Sequential Processes (CSP) formalism, the GSFC team has found debugging to be straightforward. The effectiveness of this formal specification is demonstrated in the context of errors involving race conditions.

In Chapter 11, Pecheur et al. present two important applications of formal verification to Livingstone, a model-based health monitoring and diagnosis system developed at NASA Ames. Prior to verification, an automatic translation method, described in the chapter, converts Livingstone models into specifications that can be verified with the Carnegie Mellon SMV model checker. Symbolic model checking is then performed with SMV. As a last step, diagnostic error traces are converted from SMV back to Livingstone. The two applications that are explored are the use of Livingstone for an In-Situ Propellant Production (ISPP) plant, a system intended to produce spacecraft propellant using the Martian atmosphere, and the Remote Agent architecture on the Deep Space One spacecraft. In both cases, application of the verification methodology described in the chapter resulted in identification of modeling errors. This proves the effectiveness of verification for highly complex real-world applications.

Finally, Appendix A discusses a linear time algorithm for walking and-or graphs that was referenced in Chapter 9.

The objective of this book is to provide an in-depth view of some of the key issues related to agent technology from a formal perspective. However, since this is a relatively new interdisciplinary field, there is enormous room for further growth. We hope that this book will not only create an initial foundation for some of the key issues, but will also point out many of the gaps in the field, thus indicating open problems to be addressed by future researchers and students.

Christopher A. Rouff
Michael Hinchey
James Rash
Walter Truszkowski
Diana F. Spears

Springer
http://www.springer.com/978-1-85233-947-0

Agent Technology from a Formal Perspective
Rouff, C.; Hinchey, M.; Rash, J.; Truszkowski, W.;
Gordon-Spears, D.F. (Eds.)
2006, XVI, 354 p., Hardcover
ISBN: 978-1-85233-947-0