Lecture Notes in Artificial Intelligence 1037
Subseries of Lecture Notes in Computer Science
Edited by J.G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

M. Wooldridge J.P. Müller M. Tambe (Eds.)

Intelligent Agents II

Agent Theories, Architectures, and Languages

IJCAI'95 Workshop (ATAL)
Montréal, Canada, August 19-20, 1995
Proceedings

Second printing

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon Universtiy, USA

Jörg Siekmann, University of Saarland, DFKI, Germany

Volume Editors

Michael Wooldridge
Jörg Müller
Zuno Ltd., International House
Ealing Broadway Centre, London W5 5DB, UK
E-mail: (mjw/jpm)@dlib.com

Milind Tambe
University of Southern California, Information Sciences Institute
4676 Admiralty Way, Marina Del Rey, CA 90292, USA
Email: tambe@isi.edu

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Intelligent agents II : agent theories, architectures, and languages ; proceedings / IJCAI '95 workshop (ATAL), Montreal, Canada, August 19 - 20, 1995. Michael Wooldridge ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; 1037 : Lecture notes in artificial intelligence)
ISBN 3-540-60805-2
NE: Wooldridge, Michael J. [Hrsg.]; IJCAI <14, 1995, Montreal>; ATAL <2, 1995, Montreal>; GT

CR Subject Classification (1991): I.2, D.2, C.2.4

ISBN 3-540-60805-2 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10628274 06/3111 - 5 4 3 2 1 Printed on acid-free paper

Preface

The past decade has been witness to an explosion of interest in the issues surrounding the design and implementation of agents that can make rational decisions and act autonomously in time-constrained, open, multi-agent environments. The aim of the *Agent Theories, Architectures, and Languages* (ATAL) workshop series is to bring together researchers interested in this exciting new technology. The ATAL workshops address the issues of agent specification via agent theories, the ability of agents to model other agents, and the ability of agents to make decisions in time-constrained environments. In particular, the workshops focus on the link between theories of agents and the realization of such theories using software architectures or languages.

This volume contains revised versions of twenty-six papers that were first presented at the ATAL-95 workshop, which was held at the International Joint Conference on Artificial Intelligence (IJCAI) in Montréal, Quebec, in August 1995. This was the second workshop in the ATAL series, building on the success of ATAL-94, which was held at the European Conference on Artificial Intelligence (ECAI) in August 1994. Fifty-four papers were submitted to ATAL-95, from sixteen different countries: fourteen from the USA, eight from the UK, seven from Canada, three each from Germany, Japan, Italy, France, and the Netherlands, two each from Australia and Portugal, and one each from Norway, Switzerland, Cyprus, China, Spain, and Mexico. Of these fifty-four papers, twenty-six were accepted for presentation and subsequent publication in this volume. It is both our hope and our expectation that this volume will serve as a useful and informative guide to the ongoing development of agent technology.

November 1995

Michael Wooldridge,
Jörg P. Müller, and
Milind Tambe

Workshop Organization

Organising Committee

Michael Wooldridge (CHAIR)	(Manchester Metropolitan University, UK)
Klaus Fischer	(DFKI Saarbrücken, Germany)
Piotr Gmytrasiewicz	(University of Texas at Arlington, USA)
Nicholas R. Jennings	(Queen Mary & Westfield College, UK)
Jörg P. Müller	(DFKI Saarbrücken, Germany)
Milind Tambe	(ISI — University of Southern California, USA)

Program Committee

Michael Beetz	(USA)	Cristiano Castelfranchi	(Italy)
Phil Cohen	(USA)	Ed Durfee	(USA)
Tim Finin	(USA)	Michael Fisher	(UK)
John Fox	(UK)	Michael Georgeff	(Australia)
Hans Haugeneder	(Germany)	Joachim Hertzberg	(Germany)
Lewis Johnson	(USA)	Sarit Kraus	(Israel)
Anand Rao	(Australia)	Charles Rich	(USA)
Jeff Rosenschein	(Israel)	John Self	(UK)
Yoav Shoham	(USA)	Candace Sidner	(USA)
Munindar Singh	(USA)	Donald Steiner	(Germany)
Kurt Sundermeyer	(Germany)	Michael Wellman	(USA)

Additional Reviewers

Alastair Burt	Susanne Kalenka
David Kinny	Michael Kolb
Hans-Jürgen Ohlbach	Simon Parsons
Markus Pischel	Michael Thiel
Gil Tidhar	Elizabeth Sonenberg

Acknowledgments

We would like to thank AAAI/IJCAI for supporting ATAL-95, and in particular, Carol Hamilton and her staff for their excellent run-time help. The editors would like to stress that the organization of ATAL-95 was a six-person job: although only three names appear as editors of this book, the workshop could not have been such a success without the assistance of Klaus Fischer, Nick Jennings, and Piotr Gmytrasiewicz. All the organizers would like to thank Joseph Bates, who gave an excellent invited presentation at the workshop. Unfortunately, the tight publication deadline prevented Joe from writing an article for inclusion in this volume. Finally, the organization of this workshop was made possible by financial support from the Department of Computing at Manchester Metropolitan University.

Contents

Introduction

This book is structured into seven sections, reflecting most current major directions in agent-related research. The first section, entitled *Theories*, contains seven papers describing (mostly logic-based) formalisms for the explanation, analysis, or specification of automated agents. The second section, *Agent Capabilities*, provides work that focuses on models of specific components or functionalities of agents. The third section, *Agent Modelling*, describes four different approaches to modelling other agents' behaviour or mental states from the perspective of an individual agent. The next section, on *Agent Architectures*, focuses on the integration of different components and functionalities into a coherent control framework for an individual agent. In contrast, the section on *Multi-Agent Architectures* is comprised of methodologies and architectures for groups or networks of agents, consisting of models for the individual agent as well as of organizational models. The sixth section, *Languages*, contains four papers that describe work on agent-based programming languages. The papers in the final section, titled *Agent Testbeds and Evaluation*, are examples of research aimed at providing testbeds for evaluating alternatives in agent design.

Clearly, these seven sections are by no means orthogonal. For example, most papers that describe (multi-)agent architectures contain links to formalisms and languages for their operationalisation, and most of them also provide their testbeds and evaluation tools. Thus, our classification is only intended as a guide to the reader interested in different aspects of agent technology.

Part I: Theories

Krogh — agents as legal entities: Christen Krogh [18] discusses some philosophical and legal issues associated with the notion of intelligent computer agents. His starting point is the observation that it may be sensible to treat agents as 'legal entities' in the sense that organizations are treated as such within, for example, English law. If (as seems likely) agent technology becomes well established, then eventually the legal status of computer agents must come into question. As legal entities, agents will have placed upon them certain obligations, and may expect to enjoy certain rights. Krogh examines some issues surrounding the treatment of agents as legal entities using, as a formal tool, a *deontic logic* (i.e., a logic of obligation [23]) that contains modalities for *seeing to it that* (stit) [2]. The stit modality used by Krogh is not given a formal (model theoretic) semantics, but the intuition is similar to the stit modality used by Wooldridge in his agent specification language [43].

van Linder et al. — preferences, goals, and commitments: In [41], van Linder *et al.* present an integrated agent logic, containing modalities for knowledge, belief, and ability. This logic is used to formalize *motivational* attitudes of agents. First, a unary modal connective $\mathbf{P}_i$ is defined to represent *preference* (thus preference is not a binary relation: one does not prefer x to y). The semantics of preference is given using a possible-worlds framework, adapted to prevent the problems of logical omniscience using an 'awareness filter' similar to that proposed for doxastic logics by Fagin and Halpern [9]. Goals are then defined to be unfulfilled 'realistic' preferences. Realism is defined as there being some course of action open to the agent that might achieve the goal. A commitment modality is then introduced, and the relationship between commitments, ability, knowledge, and goals is discussed.

Rao — decision procedures for BDI logics: BDI agents are systems whose internal state may be characterized in terms of 'mental states' loosely corresponding to beliefs, desires, and intentions. Anand Rao was one of the earliest to recognize the significance of the BDI model, and has, in a series of joint papers with Michael Georgeff, formalized and implemented many aspects of this model (see, e.g., [32, 33, 34]). Much of this work has been based on branching-time logics that contain modalities for beliefs, desires, and intentions. In [31], Rao considers proof methods for such logics. He gives decision procedures for linear-time variants of the BDI logics presented in previous work, and uses these decision procedures to prove completeness for axiomatisations of the logics.

Singh — an agent specification logic: Munindar Singh [37] develops an agent specification logic containing modalities for knowledge, intention, and 'know-how'. This logic builds on his prior work in agent theory [36]. The underlying model of action and time is based on a strict partial order of temporal moments, denoting different world states. Actions effect transitions between moments; they determine the granularity at which agents can make choices. Singh draws a distinction between *know-that* and *know-how*. The former category of knowledge characterises the states among which an agent cannot distinguish. Know-how characterises what world states an agent can force to occur. The intuitive meaning of intentions is that of qualitative preference of courses of events by the agent. Intentions are the conditions that inevitably hold on each of the courses of events selected by the agent.

Traverso et al. — reasoning with failure: In [40], Paolo Traverso and colleagues observe that much work in agent theory (and indeed, AI generally), ignores the possibility of *failure*. It is generally assumed, for example, that if an agent has a plan to achieve ϕ, then that plan will succeed. Thus, effort is generally directed at algorithms for *constructing* plans, under the assumption that such plans will succeed. Of course, the real world is not so benevolent: an agent's plans frequently fail, either through the interference of other agents, or else by actions simply failing to have the intended effects. Traverso and colleagues develop a variant of process logic that contains a program constructor `iffail`, which allows the development of plans that specify what to do in the case of actions failing. That is, it allows one to specify an agent action `iffail` α `then` β `else` γ, the intended semantics of which are 'do α, and if α fails, do β, otherwise do γ'. The logic is then extended by the introduction of an operator $Sensed(c)$ to represent an agent sensing the information c.

Wooldridge — time, knowledge, and choice: In [43], Wooldridge considers the relationship between agent theories, and the architectures to realise such theories. He develops a specification language for agents, based (like Singh's logic [37] and earlier work by Rao and Georgeff [32, 33, 34]) on the expressive branching time logic CTL* [8]. In addition to the CTL* branching time connectives, Wooldridge's logic contains modal connectives for representing *knowledge* (information implicit within an agent's state), and *seeing to it that* (stit). The logic allows one to represent statements that capture the sense of 'if the agent ever knows ϕ, then it should see to it that ψ'. However, unlike many other formalisms, the semantics of the knowledge and stit modalities in Wooldridge's logic are *grounded* — given a concrete interpretation in terms of the states and actions of automata (cf. [35, 15]). This allows Wooldridge to establish a clear relationship between the agent specification language and automata that realise the specification: given a model M that satisfies a specification ϕ, Wooldridge shows how one can extract from M an automaton that realises ϕ.

Part II: Agent Capabilities

Hexmoor — learning from routines: Henry Hexmoor [16] focuses on the problem of enabling an agent to learn from routine interactions with its world. In particular, an agent, starting out with no *a priori* preferences for a course of action, is to learn preferences for all of the choice points it faces via interactions with its world. The key to Hexmoor's approach is the concept of a *goal sketch*, a knowledge structure that encodes an agent's understanding of a goal. This structure forms the basis of a reward system in an agent's routine task performance, e.g., a goal sketch would yield an agent rewards for accidentally stumbling upon a goal. Hexmoor has implemented this learning approach in a Lego robot called Garry, based on a three-layered architecture called GLAIR (see also [5] and [12] for other layered architectures). Garry learns to move about on a black-and-white table-top to reach its reward, a beacon.

Chu-Carroll and Carberry — collaborative planning: Jennifer Chu-Carroll and Sandra Carberry [6] focus on multi-agent collaborative planning, that is captured by a recursive *propose-evaluate-modify* framework of actions. In this framework, an agent engaged in collaborative planning proposes modifications to the agents' *sharedplan* under development, while the other agent evaluates and suggests modification to this proposal if unacceptable. This process then recurs. Proposing a modification itself presents two challenges: (i) selecting a focus of modification; and (ii) for the chosen focus of modification, selecting appropriate evidence to justify the modification. The authors present several heuristics to address these issues, and back these up with examples of collaborative negotiation.

Part III: Agent Modelling

Bicchieri et al. — games servers play: Cristina Bicchieri, Eithan Ephrati, and Aldo Antonelli [4] discuss the design of an effective procedure for multi-agent coordination within a game-theoretic framework. The authors argue that while game theory provides powerful tools for modelling agent interactions, it does not provide inference procedures to enable an agent to reason to a solution. The procedure the authors develop is based on common knowledge of rationality among agents. By modelling other agents as rational entities, and assuming such rationality to be common knowledge, agents can converge to an equilibrium point. In particular, an agent rules out inferior strategies, and recursively reasons that other agents will do the same (recursive modelling is also discussed in [14, 42]). This procedure has some similarity to the iterated elimination of dominated strategies in strategic form games.

Gmytrasiewicz — the recursive modelling method: Piotr Gmytrasiewicz [14] discusses some issues in agent modelling, and in particular, recursive modelling. He focuses on an influential strand of philosophical thought: Daniel Dennett's *ladder of agenthood*, and *stance* (intentional, physical or design stance). He examines the close relationship between these ideas and the recursive modelling method (RMM), a decision-theoretic framework that allows an agent to recursively reason about other agents, and to pick out a utility maximizing action within this context. This decision-theoretic model of reasoning is based on an agent's subjective perspective, and it is contrasted with a traditional game-theoretic approach, which takes more of a global, objective perspective.

Tambe and Rosenbloom — agent tracking: Milind Tambe and Paul Rosenbloom [39] analyse the requirements for *agent tracking* — dynamic modelling of other agents' higher-level desires and intentions, based on observations of their actions — for tasks in a real-world, dynamic, multi-agent environment. Based on this analysis, they recommend that agent architectures should support several capabilities to facilitate agent tracking. One key recommendation is that an agent architecture, in addition to generating the agent's own behaviour, needs to execute runnable models of other agents, to simulate their behaviour. Other recommendations focus on facilitating real-time needs of such an architecture. The authors present an implementation of an architecture — a variant of the Soar architecture — based on these recommendations, and use that to implement pilot agents in a real-world synthetic combat environment.

Vidal and Durfee — agent modelling using limited rationality: José Vidal and Edmund Durfee [42] argue that when reasoning with recursively nested models of other agents, a resource-bounded agent must selectively prune away less useful models, or else it may be overwhelmed by the cost of reasoning. They present an approach, that is also based on RMM [14], to aid an agent in such selective pruning of less important models. The key to such selectivity is the notion of *expected gain*: an agent should deliberate over a recursively nested model only if such deliberation would lead to a change in its future course of actions. This approach is implemented on a well-known multi-agent problem, *the pursuit task*, which involves predators attempting to surround a prey. Experimental results are presented that indicate the benefits of the authors' approach.

Part IV: Agent Architectures

Bonasso et al. — 3T: [5] describes a layered control architecture for autonomous robots. It consists of three control layers, a reactive skill layer, a sequencing layer, and a deliberation layer. The reactive skill layer provides a set of situated skills. Skills are capabilities that, if placed in the proper context, achieve or maintain particular states in the world. The sequencing layer is based on the RAPs system [10, 11]. It maintains routine tasks that the agent has to accomplish. The sequencing layer triggers continuous control processes by activating and deactivating reactive skills. Finally, the deliberation layer provides a deliberative planning capability which selects appropriate RAPs to achieve complex tasks. This selection process may involve reasoning about goals, resources, and timing constraints. A further interesting aspect of the paper is the discussion of different dimensions to provide assistance to the designer of an agent system in deciding whether a specific aspect of a task belongs at the skill level, at the sequence level, or at the deliberation level. A set of tools is presented to help design robot applications, and descriptions of different robot systems are given that have been implemented using the 3T architecture.

Fischer et al. — INTERRAP*:* Klaus Fischer and colleagues [12] describe a redesign of the INTERRAP control architecture [27, 28] according to the Belief-Desire-Intentions (BDI) paradigm. Similar to 3T [5], INTERRAP has three interacting control layers: the behaviour-based layer, the local planning layer, and the cooperative planning layer. Each layer instantiates a generic structure providing five functionalities: situation recognition, goal activation, planning, scheduling, and execution. The behaviour-based layer accounts for the reactivity of the agent and for the scheduling of routine tasks (procedures). Thus, it contains capabilities of the reactive skill layer and (partly) of the sequencing layer of 3T. In contrast to the other architectures presented in this volume, a distinction is drawn in INTERRAP between the local planning of tasks, which is achieved by the local planning layer, and multi-agent planning, which is realised by the cooperative planning layer. Thus, INTERRAP explicitly supports the design of agents for cooperative, multi-agent environments.

Norman and Long — motivated agency: In [29], Norman and Long investigate two issues: a framework for building *motivated agents*, and the control of agents' reasoning attention. In their framework, an agent is driven by a set of motives that may lead to goal generation. However, an agent capable of generating its own goals may end up with a large number of goals, beyond its resource capacity. Thus, it must limit its attention to the most salient goals, to achieve them in real-time. A heuristic mechanism called *alarms* is presented that achieves such attention focusing. An alarm is a function that associates a goal with a function of intensity over time, where intensity reflects the appropriateness of the goal to the current situation. An agent will bring a goal into its attentional focus if that goal's intensity increases above a pre-specified threshold. The authors claim that such attentional focusing concentrates planning and reasoning effort, and avoids unnecessary reasoning.

Part V: Multi-Agent Architectures

Barbuceanu and Fox — agent building shell: In [1], the authors describe a shell for building agent systems, which offers several reusable layers of languages and services supporting agent-based software engineering. To describe the structure of the individual agent, the authors define a layered architecture (see also [5, 12, 38]) consisting of a knowledge management component, a coordination model, a communication layer, an application interface, and modules for information distribution and conflict management which both work on an organizational model. A terminological knowledge representation system is used to maintain the agent's beliefs. Information distribution is done based on the content of the information and on a representation of other agents' interests. Coordination protocols are described in the coordination language COOL which, like KQML [21], is based on the notion of speech acts or performatives. Conflict management addresses the resolution of incompatibility of the beliefs of different agents by a model based on the notions of *credibility* and *deniability* of agents, where credibility provides an ordering of agents with respect to specific roles, and deniability defines the cost of retracting a specific piece of information. The approach has been applied to the domain of integrated supply chain management.

Iglesias et al. — MIX: MIX [17] is an architecture for the design of multi-agent systems that allow an integration of both connectionist and symbolic subsystems. MIX consists of two sub-models: the *agent model* describing the structure of an individual agent, and the *network model* defining the communication infrastructure used by agents. A MIX agent consists of a control module, a database, and an interface to the communication network. Agents are implemented using MIX-ADL, an object-oriented, declarative agent specification language. The network model provides the agents with a uniform view of the network. A distinction is drawn between agents offering network-related services (so-called network agents) and application agents. The network model offers various facilities like a yellow-page service (comparable to the facilitator in [30]), coordination facilities (e.g., the contract net protocol [7]), and rudimentary knowledge facilities supporting communication among heterogeneous information agents (see also [21]).

Pelletier and Arcand — CBA: The Cognitive-Based Architecture (CBA) [30] is an architecture for the design of systems of agents that are distributed over different locations, that are communicating via a telecommunications network using one from a set of communication languages or protocols, and that should be able to form work-groups dynamically (see [1, 17] for examples of related approaches within this volume). CBA offers three different types of agents: junior intelligent agents (JIAs), which are specialists in a particular domain; senior intelligent agents (SIAs), which maintain a long-term evolutionary knowledge base containing experiences and share this knowledge with JIAs; and a facilitator agent (FA), which provides important system functionalities like optimizing the communications network, brokering of services, and providing registration services for JIAs and SIAs. Work-groups are formed dynamically upon request from certain participants.

Mullen and Wellman — market-oriented agents: Tracy Mullen and Michael Wellman [25] focus on the framework of market-oriented agents, where agents participate in a computational market economy, and interact with other agents via computational goods and services. This perspective has the benefit of bringing to bear analytical tools from economic theory, both for building individual agents and predicting their aggregate behaviour. The authors point out that scaling up from small-scale, static market environments to large-scale, dynamic ones raises some challenging issues. For instance, in static environments, agents know all the goods needed ahead of time, but in dynamic environments, the goods and their distinctions change over time. Advances in agent theories may aid in addressing such issues.

Part VI: Languages

Beyssade et al. — agents in higher-order logic: In [3], Beyssade *et al.* present a framework for agent theory in which agents are represented using higher-order logic. The formalism allows one to represent heterogeneous agents, (i.e., agents that have different internal knowledge representation languages), that communicate via message passing. The specific examples of KR languages given in the paper are based on typed λ-calculus. The reasoning ability of agents is represented via theorem proving.

Giroux — ReActalk: ReActalk [13] is a platform for the design of and the experimentation with heterogeneous agents in open and inter-operable environments. Agents in ReActalk are *reflective*, i.e., they can reason or act upon their own behaviour. ReActalk primarily addresses the need for agents to adapt their behaviour to changes within the environment. Adaptation can be achieved by either modifying the behaviour of the agent itself, or by modifying the agent's environment. ReActalk offers various mechanisms of adaptation through self-modification ensuring inter-operability between different agents using different models of computation, such as the acquisition and the removal of somatic capacities.

Lespérance et al. — CONGOLOG: In [19], Lespérance *et al.* present CONGOLOG, an agent-oriented extension of their GOLOG logic programming language. The GOLOG language, which has been implemented in PROLOG, is based on the situation calculus [22], and draws on Moore's theory of knowledge and ability [24]. In CONGOLOG, agents communicate via message passing; each agent has associated with it a message queue, which it can access via message-read actions. The messages that agents send look very much like KQML performatives [21], with a logical semantics defined within the language. The authors show how CONGOLOG can be used to specify a meeting scheduling system.

Mayfield et al. — KQML: Mayfield, Labrou, and Finin [21] present a list of desiderata for languages and protocols for communication among information agents. These desiderata include the form, the content, the semantics, the implementation, the appropriateness for networking, the ability to fit distributed, heterogeneous, and dynamic environments, and the reliability of the languages and protocols, respectively. The authors then describe the Knowledge Query and Manipulation Language (KQML), which was developed within the *External Interfaces Group* of the ARPA Knowledge Sharing Effort. The aim of KQML is to provide a standard for communication among intelligent information agents in decentralised, heterogeneous, and dynamic environments. Different features of KQML are presented, and the language is evaluated relative to the desiderata that were specified. The main result of the evaluation is that, while KQML corresponds well to most of the desiderata regarding form, content, implementation, networking, and environment, the semantics of KQML as well as security and authentication (which contribute to the reliability of the system) are still open issues.

Part VII: Agent Testbeds and Evaluation

Malheiro and Oliveira — multi-agent belief revision: Benedita Malheiro and Eugénio Oliveira [20] discuss their design of a testbed for distributed belief revision. The testbed consists of a society of communicating agents, where each agent maintains a belief space partitioned into two sets: (i) private beliefs, which the agent keeps to itself; and (ii) shared beliefs, shared with at least one other agent. The authors' goal is to enable a user to experiment with several belief revision policies for such a multi-agent world. These policies include: (i) a pure *local consistency* mode, where an agent maintains consistency among facts that it originates; (ii) an *inconsistency sharing* mode, which adds the requirement that inconsistencies among shared facts be communicated to relevant agents, and (iii) a *shared global consistency* mode, which adds the requirement of global consistency among shared facts.

Müller — evaluation of behaviour-based decision-making: Jörg Müller [26] investigates how the local design of the decision-making of agents influences the global performance of a multi-agent system. He provides a predictive model based on Markov chain theory for a specific class of local agent behaviours, namely those based on behaviour-based decision-making as it is realised by the behaviour-based layer of the INTERRAP architecture [12]. This model is based on Markov chain theory. By the example of robot navigation tasks, the model is used to show that the use of intuitive local decision methods that lead to quasi-optimal performance in the single-agent case may result in pathological behaviour if different agents using these methods co-exist in a multi-agent environment. The theoretical model is complemented by empirical results [12] which show that the system behaviour can be drastically improved by allowing explicit cooperation among agents through communication.

Sloman and Poli — SIM_AGENT: SIM_AGENT is a toolkit that allows an agent developer to explore different options in the design of autonomous agents. The underlying agent architecture consists of a set of complex interacting processes, each of which accounts for a particular class of agent capabilities. The most basic class of processes are *reflexes*, which can be compared to the reactive skills in 3T [5] and to the reactor patterns of behaviour in INTERRAP [12]. Others are automatic (pre-attentive) processes, which are similar to the procedure patterns of behaviour in INTERRAP, and to the structures maintained at the sequencing layer of 3T. The third class of processes are reflective management processes, like planning, scheduling, and deciding. These processes, which are inherently resource-bounded, roughly correspond to the deliberation layer in 3T and to the local and cooperative planning layers of INTERRAP. In contrast to the other architectures presented in this volume, Sloman and Poli's architecture provides a separate modelling of meta-management processes that regulate, direct, and monitor the management processes. In 3T and INTERRAP, these processes are realised inside the respective layers.

November 1995

Michael Wooldridge,
Jörg P. Müller, and
Milind Tambe

References

1. M. Barbuceanu and M. S. Fox. The architecture of an agent building shell. (In this volume).
2. N. Belnap and M. Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, 1988.
3. C. Beyssade, P. Enjalbert, and C. Lefèvre. Cooperating logical agents: Model, applications. (In this volume).
4. C. Bicchieri, E. Ephrati, and A. Antonelli. Games servers play: A procedural approach. (In this volume).
5. R. P. Bonasso, D. Kortenkamp, D. P. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive agents. (In this volume).
6. J. Chu-Carroll and S. Carberry. Conflict detection and resolution in collaborative planning. (In this volume).
7. R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63 – 109, 1983.
8. E. A. Emerson and J. Y. Halpern. 'Sometimes' and 'not never' revisited: on branching time versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.
9. R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490, Los Angeles, CA, 1985.
10. R. James Firby. *Adaptive Execution in Dynamic Domains.* PhD thesis, Yale University, Computer Science Department, 1989. Also published as Technical Report YALEU/CSD/RR#672.
11. R. James Firby. Task networks for controlling continuous processes. In *Proceedings of the 1994 Conference on AI Planning Systems*, pages 49–54, 1994.
12. K. Fischer, J. P. Müller, and M. Pischel. A pragmatic BDI architecture. (In this volume).
13. S. Giroux. Open reflective agents. (In this volume).
14. P. J. Gmytrasiewicz. On reasoning about other agents. (In this volume).
15. J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2:37–68, 1987.
16. H. H. Hexmoor. Learning from routines. (In this volume).
17. C. A. Iglesias, J. C. González, and J. R. Velasco. MIX: A general purpose multiagent architecture. (In this volume).
18. C. Krogh. The rights of agents. (In this volume).
19. Y. Lésperance, H. J. Levesque, F. Lin, D. Marcu, R. Reiter, and R. B. Scherl. Foundations of a logical approach to agent programming. (In this volume).
20. B. Malheiro and E. Oliveira. Consistency and context management in a multi-agent belief revision testbed. (In this volume).
21. J. Mayfield, Y. Labrou, and T. Finin. Evaluating KQML as an agent communication language. (In this volume).
22. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*. Edinburgh University Press, 1969.
23. J. J. Ch. Meyer and R. J. Wieringa, editors. *Deontic Logic in Computer Science — Normative System Specification.* John Wiley & Sons, 1993.
24. R. C. Moore. A formal theory of knowledge and action. In J. F. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers: San Mateo, CA, 1990.
25. T. Mullen and M. P. Wellman. Some issues in the design of market-oriented agents. (In this volume).

26. J. P. Müller. A markovian model for interaction among behavior-based agents. (In this volume).
27. J. P. Müller and M. Pischel. An architecture for dynamically interacting agents. *International Journal of Intelligent and Cooperative Information Systems (IJICIS)*, 3(1):25–45, 1994.
28. J. P. Müller and M. Pischel. Integrating agent interaction into a planner-reactor architecture. In M. Klein, editor, *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*, Seattle, WA, USA, July 1994.
29. T. J. Norman and D. Long. Alarms: An implementation of motivated agency. (In this volume).
30. S.-J. Pelletier and J.-F. Arcand. Cognitive based multiagent architecture. (In this volume).
31. A. S. Rao. Decision procedures for propositional linear-time Belief-Desire-Intention logics. (In this volume).
32. A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473–484. Morgan Kaufmann Publishers: San Mateo, CA, April 1991.
33. A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 439–449, 1992.
34. A. S. Rao and M. P. Georgeff. A model-theoretic approach to the verification of situated reasoning systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 318–324, Chambéry, France, 1993.
35. S. Rosenschein and L. P. Kaelbling. The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 83–98. Morgan Kaufmann Publishers: San Mateo, CA, 1986.
36. M. P. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*. Springer-Verlag: Berlin. Lecture Notes in Artificial Intelligence Volume 799. 1994.
37. M. P. Singh. Semantical considerations on some primitives for agent specification. (In this volume).
38. A. Sloman and R. Poli. SIM_AGENT: A toolkit for exploring agent designs. (In this volume).
39. M. Tambe and P. S. Rosenbloom. Agent tracking in real-time dynamic environments. (In this volume).
40. P. Traverso, L. Spalazzi, and F. Giunchiglia. Reasoning about acting, sensing, and failure handling: A logic for agents embedded in the real world. (In this volume).
41. B. van Linder, W. van der Hoek, and J. J. Ch. Meyer. How to motivate your agents. (In this volume).
42. J. M. Vidal and E. H. Durfee. Recursive agent modeling using limited rationality. (In this volume).
43. M. Wooldridge. Time, knowledge, and choice. (In this volume).