Technical Report No: 2003/20

Using safety and liveness properties to drive learning in a multi-agent system

Robert Logie
Jon G. Hall
Kevin G. Waugh

10th December 2003

Department of Computing
Faculty of Mathematics and Computing
The Open University
Walton Hall,
Milton Keynes
MK7 6AA
United Kingdom

http://computing.open.ac.uk

Using safety and liveness properties to drive learning in a multi-agent system.

Robert Logie[1], Jon G. Hall[2], and Kevin G. Waugh[2]

[1] Osaka Gakuin University, Faculty of Informatics
2-36-1 Minami Kishibe, Suita Shi Osaka, 564-8511 Japan
rob@utc.osaka-gu.ac.jp
[2] The Open University, Department of Mathematics and Computing,Walton Hall,
Milton Keynes MK7 6BJ England

Abstract. One of the strongest results in temporal logic is Chang, Manna and Pnueli's partitioning of reactive system properties into the classes of safety and liveness[1]. Safety and liveness properties state, intuitively and respectively, that something bad will not happen and that something good will eventually happen. In this paper we show how, in a multi-agent world, this safety/liveness partitioning can be used to drive learning. If an agent is introduced to a world and given a set of descriptions of system safety an liveness properties then how is it to discover how to behave in such a way as to satisfy them? Safety and liveness properties will influence agent behaviour, safety properties are cast as system norms exerting a restraining influence whilst liveness properties are cast as desires which exert a driving influence. Agents will randomly gather a set of atomic behaviours - simple actions which may be used individually, in combination or in conjunction with other agents. In order to discover behaviours which satisfy these system properties agents must have a "mischevious" element in their behaviour. Future worlds are given a preference ordering, when this ordering fails to provide clear guidance an agent may "mischeviously" select any available action not proscribed by safety norms. Undesirable world states are described by these safety norms and agents will be obliged to prevent these states by either refraining from actions which are known to bring them about or acting so as to attempt to clear these states if they are detected. A small number of dedicated coaching agents will assist "normal" agents in refining any behaviours they have developed. Coaches will also try to ensure that successful behaviours are propagated as quickly as possible. The mechanism for achieving these combined behaviours is a novel combination of belief update and belief revision. This arrangement provides a belief management framework which is capable of identifying factors governing the behaviour of the agent's world with no requirement for prior knowledge. The resulting set of beliefs will be filtered by an agent's desires and intentions so as to produce a partially ordered set of plausible worlds and, hence, a partial order on sets of available actions to control the agent's behaviour.

1 Introduction

Jennings et al.[2] state that *"A major selling point of purely reactive systems is that overall behaviour emerges from the interaction of the component behaviours when the agent is placed in its environment."* One of our goals is to develop a mechanism which allows reactive agents to evolve towards behaving in a way that satisfies a system specification. The promise of emergent behaviour combined with their inherent simplicity makes reactive agents very attractive. There are, however, a number problems. Jennings notes that the relationship between atomic behaviours, the environment and the resulting overall behaviour may be extremely difficult to understand. Despite their relative simplicity specifying agents for particular tasks is potentially very complex. An additional problem is that it is difficult to see how to design a purely reactive agent that will learn from its experience and improve its performance over time, a satisfactory behaviour may emerge but agents may be unable to improve on this even over an extended run.

We propose to the adoption of three methods to get round these problems. Firstly a hybrid reactive/cognitive agent architecture biased towards reactive operation and secondly, the introduction of dedicated *coaching agents* to assist *general agents* in developing and propagating successful behaviours. The third element is a novel combination of belief management techniques, coaching agents will use belief update to manage their knowledge whilst general agents will use belief revision. The interaction between these methods will drive system learning.

The next section briefly outlines belief management, plausibility and related work. Section 3.3 describes how the proposed belief management system will operate and section 3.6 outlines how such a system may be accommodated by a hybrid agent architecture.

2 Belief management and plausibility

2.1 Beliefs and the BDI agent model

The role of *attitudes* such as belief, desire and intention has long been recognised in agent design[3](see [4],[5]). This section briefly introduces beliefs, desires and intentions from a philosophical background then concentrates on beliefs to examine two approaches to belief management.

An agent's knowledge and beliefs may be described as *information attitudes*[6] and its desires and intentions as *pro-attitudes*[4]. Pro-attitudes have an influence over how an agent behaves. If I feel uncomortably hot then I may desire to be a feel cooler. At the same time I believe that the air conditioner on will make me cooler and form the intention to turn the air conditioner on. Desires and intentions differ in that intentions are conduct controlling pro-attitudes whereas desires are potential influencers of action. An intention, thus, involves a commitment to action that is absent from a desire, Bratman terms this relation between intention and action the *volitional dimension* of commitment.

The simple example above uses a belief-desire model, my intention to turn the air conditioner on can be reduced to the belief that the air conditioner will make me feel cooler and a desire to feel cooler. This model has both descriptive and normative aspects, it attempts to structure a commonsense approach to action and to articulate a practical rationality. If one accepts the existence of a predominant desire then this model is adequate. My having a predominant desire to become cooler means that I desire this more than performing any other option that I deem incompatible. My intention to turn the air conditioner on cannot be identified with such a predominant desire because it does not admit either a volitional or reasoning centred commitment.

The tactic of reducing intentions to beliefs and desires is, it would appear, inadequate. Bratman goes on to consider intentions in the context of a boundedly rational agent. Bounded rationality introduces a requirement for *practical* as opposed to omniscient reasoning. Bratman frames intentions as partial plans which can play a role in future reasoning. Intentions are both inputs to and outputs from an agent's reasoning. As inputs they can pose problems - I want to turn the air conditioner on but where is the remote controller. They can also pose constraints - my desire use the air conditioner may mean that I cannot use certain other electrical apparatus. Intention has two facets, one deals with intentional action and the other with coordinating plans, recognising these facets forces us to consider intention as a distinct element of agency.

Cohen and Levesque[7] formalised Bratman's work and identified seven properties forming what Wooldridge terms *a desiderata for intention*. These are listed in full here although we are concerned mainly with properties 3, 4, 5 and 6 - those which are pertinent to belief management. Properties 1 and 2 correspond to with Bratman's *reasoning centred dimension* of commitment.

Property 1 *Intentions pose problems for agents who then need to determine how to solve them.*

Property 2 *Present intentions can "filter" future intentions so as to prevent the adoption of conflicting intentions.*

Property 3 *Agents monitor the success of their intentions, if attempts at a particular intention fail then an agent will try again.*

Property 4 *Agents believe that their intentions are possible. It would be irrational to adopt an intention believed impossible.*

Property 5 *Agents do not believe that they will not bring about their intentions.*

Property 6 *Agents believe that they will bring about their intentions in certain circumstances.*

Property 7 *Agents need not intend all of the side effects, expected or otherwise, of their intentions.*

2.2 Belief management

Beliefs are a core element of a rational agent's behaviour, properties 4, Properties 5 and Properties 6 outline the influence that beliefs exert over the adoption of intentions. An agent newly placed in a world may, initially, have a very small set of beliefs about its surroundings. As the agent discovers new facts or properties of its world then it will need to change its set of beliefs. Understanding how an agent ought to change its beliefs given new information has been an active research area in both philosophy and AI. Two approaches to this change have been studied in detail, *belief revision* and *belief update*.

Belief revision Belief revision originated from studies in the philosophy of science and occurs when an agent changes its set of beliefs following the adoption of a new belief. If an agent learns φ and φ is consistent with its existing beliefs then the new belief, φ, is simply added to the agent's knowledge base. If, however, φ is inconsistent with existing beliefs then the existing beliefs are revised by discarding some older beliefs so as to maintain consistency. The most commonly accepted approach to belief revision is known as the AGM theory (after Alchourrón et al,(1985); Gärdenfors, (1988), see [8]). This theory assumes that the agent's epistemic state is represented by a set K of formulas in some logical language, $\mathcal{L}_e$ over a set of primitive propositions. Belief revision takes a set of beliefs, A, a revision operator $\circ$ and a new formula, φ returning a new belief set, $A \circ \varphi$, after the operation. Intuitively this process should result in a *minimal change* to the existing belief set, Friedman and Halpern[8] list the following postulates (where Cl represents the deductive closure of the set of formulas A) to characterise this notion.

BR1. $A \circ \varphi$ is a belief set
BR2. $\varphi \in A \circ \varphi$
BR3. $A \circ \varphi \subseteq Cl(A \cup \{\varphi\})$
BR4. If $\neg\varphi \notin A$ then $Cl(A \cup \{\varphi\}) \subseteq A \circ \varphi$
BR5. $A \circ \varphi = Cl(false)$ iff $\vdash_{\mathcal{L}_e} \neg\varphi$
BR6. If $\vdash_{\mathcal{L}_e} \varphi \Leftrightarrow \psi$ then $A \circ \varphi = A \circ \psi$
BR7. $A \circ (\varphi \wedge \psi) \subseteq Cl(A \circ \varphi \cup \{\psi\})$
BR8. If $\neg\varphi \notin A \circ \varphi$ then $Cl(A \circ \varphi \cup \{\psi\}) \subseteq A \circ (\varphi \wedge \psi)$

BR1 and BR2 state that after a revision by φ the belief set should include φ. BR3 and BR4 describe *persistence* indicating that if a new belief is consistent with the existing belief set then revision should not remove any existing beliefs and should not add anything beyond that implied by the existing set and the new belief. Coherence is described by BR5 and BR6,BR5 states that the agent is capable of incorporating any consistent belief and BR6 that the syntactic form of the new belief does not affect the revision. BR7 and BR8 place constraints requiring that revision maintain belief coherency.

Belief update Belief update originated from work in the database community (see [8]) and addresses the problem of changing a knowledge base so as accommodate new facts or beliefs about the world. If an agent makes a new observation that contradicts or is inconsistent with existing beliefs then existing beliefs are not necessarily considered as being false. Belief revision does not assume that the world is static and attempts to capture changes in the world. Katsuno and Mendelzon (see [8]) list a set of postulates addressing formulas, μ, rather than belief sets which an update operation, $\diamond$, ought to satisfy.

BU1. $\vdash_{\mathcal{L}_e} \mu \diamond \varphi \Rightarrow \varphi$
BU2. If $\vdash_{\mathcal{L}_e} \mu \Rightarrow \varphi$,then $\vdash_{\mathcal{L}_e} \mu \diamond \varphi \Leftrightarrow \varphi$
BU3. $\vdash_{\mathcal{L}_e} \neg\mu \diamond \varphi$iff $\vdash_{\mathcal{L}_e} \neg\mu$ or $\vdash_{\mathcal{L}_e} \neg\varphi$
BU4. If $\vdash_{\mathcal{L}_e} \mu_1 \Leftrightarrow \mu_2$ and $\vdash_{\mathcal{L}_e} \varphi_1 \Leftrightarrow \varphi_2$ then $\vdash_{\mathcal{L}_e} \mu_1 \diamond \varphi_1 \Leftrightarrow \mu_1 \diamond \varphi_1$
BU5. $\vdash_{\mathcal{L}_e} (\mu \diamond \varphi) \wedge \psi \Rightarrow \mu \diamond (\varphi \wedge \psi)$
BU6. If $\vdash_{\mathcal{L}_e} \mu_1 \Rightarrow \mu_2$ and $\vdash_{\mathcal{L}_e} \mu_2 \Leftrightarrow \mu_1$, then $\vdash_{\mathcal{L}_e} \mu \diamond \varphi_1 \Leftrightarrow \mu \diamond \varphi_2$
BU7. If μ is complete then $\vdash_{\mathcal{L}_e} (\mu \diamond \varphi_1) \wedge (\mu \diamond \varphi_2) \Rightarrow \mu \diamond (\varphi_1 \vee \varphi_2)$
BU8. $\vdash_{\mathcal{L}_e} (\mu_1 \vee \mu_2) \diamond \varphi_1 \Leftrightarrow (\mu_1 \diamond \varphi) \vee (\mu_2 \diamond \varphi)$

This set of postulates has a number of similarities to the belief revision postulates, BU1 (c.f BR2) states that when an agent learns φ then it believes φ. BU2 (weaker than BR3 and BR4) specifies that if φ is already believed then updating an agent's beliefs with φ does not change the agent's beliefs. BU3 and BU4 (c.f. BR5 and BR6) impose coherence on any belief change process

2.3 Plausibility and agents of change

Assessing the probability of future worlds sits uneasily with the idea of systems of autonomous agents. Agents are autonomous entities which can influence their own destinies. Saying that "it is probable that two chemicals will be mixed in a future world" really means that this future world is plausible and that there is a probability that the agent will cause the chemicals to be brought together. There is certainly an element of probability but that is only part of the consideration and is heavily dependent on the agent's choice of actions.

Friedman and Halpern[9] note that future worlds can be considered in terms of plausibility and that plausibility space is a direct generalisation of probability space. Rather than mapping worlds to probabilities in $[0, 1]$ plausibility space maps worlds onto an arbitrary partially ordered plausibility set. Plausibility is an unstructured approach to uncertainty[10] which leaves agents free to decide what criteria are important to it and order worlds accordingly. Moreover, this generalisation can also be applied to other methods of dealing with uncertainty and can be easily incorporated into the Kripke semantics that are usual for dealing with possible worlds and branching time. If an agent knows little or nothing about its environment then, obviously, it is not in a position to rationally choose an action available to it. At some point the agent may come to possess sufficient knowledge to be able to do this but below this threshold agents will use their actions randomly and "mischeviously". Agents will use a modular architecture which allows them to "collect" actions, when the world is initialised a number of actions will be distributed randomly and agents will be able to collect these as they move about their world. agents will build sets available actions randomly giving the agent population a degree of diversity.

3 The agents

General agents and *coaching* agents have already been mentioned, this section briefly describes these entities and the reasons for a hybrid architecture.

3.1 General agents

Emergent behaviour is something normally associated with reactive agents and learning normally associated with cognitive agents. Both are required here. The attraction of a completely new architecture hides a number of pitfalls[11, page 235], it was decided to pick the required features from existing architectures to form a modular *hybrid* agent combining both reactive and cognitive elements. There is a body of literature on hybrid architectures covering areas such as process management agents[12], hybrid control of robots using an interface agent [13], and symbolic/reactive hybrid robot control [14]. The only novel aspects of the proposed architecture is the ability of the cognitive module to provide low level assistance to the basic agent module by analysing data from a history module which maintains a short history of the agent's internal state. Low level assistance will be provided when the reactive *core module* is unable to select an appropriate action. Agents will be internally biased to prefer reactive or cognitive operation, this means that the same architecture can be used for general and coaching agents, the only real differences being the operational bias and the resources available.

3.2 Coaching agents

Coaching agents have three main purposes, to mediate in agent interactions so as to enforce consistency on any terms used to describe world objects and events, to assist agents in identifying and refining "good" behaviours and to assist agents in discovering the properties of their environment.

When two agents communicate there may be difficulties in them being able to relate to each other. Each agent will have its own representation of the local environment and they may use different representations or names for the same actions or objects. It may be obvious to an observer that the two agents are in an identical situation but not to the agents. Agents may co-operate to achieve something by chance and be unaware of the key events in their recent histories that have got them to the point where they are co-operating. At some point in the future either of these agents may encounter a coach and give that coach their historical data, the coaching agent will be able to analyse the data from an observer's point of view and condense the important points from the behaviour history so as to produce a plan or behaviour for the joint action. Similarly if a solo agent presents a plan to a coach then that coach - using its greater stored knowledge - may identify actions or steps that are unnecessary and refine the plan for the agent. Coaching agents and general agents will use different systems for belief management, this should assist them in discovering the properties of their environment, the interaction leading to this is described in section 3.3.

Coaching agents are common in other environments, RoboCup provides many instances of coaches (see, for example, [15]) but these generally have global knowledge or a global view of the environment. Coaching agents are situated in the environment along with other agents and work with local knowledge and local perceptions. These coaches can, however, draw on the knowledge and perceptions of other agents and apply greater resources to reasoning with and about it. This gives the system two levels of reasoning, weak in general agents and strong in coaches.

3.3 Combining belief revision and belief update

Friedman and Halpern's work seeks to provide a general framework for belief management tachniques such as belief update and belief revision This work seeks to exploit the differences between them. Belief revision and belief update have generally been considered as suitable for static and dynamic worlds respectively. Friedman and Halpern[8] indicate that is more accurate to say that belief update is more suited to worlds where the *propositions used to describe the world* are static. Bringing time into the world description mean that a dynamic world can be represented by a sequence of static propositions. This is not really suitable for a reactive system which is generally accepted as maintaining an ongoing interaction with its environment[1]. Waiting for a future event is not really *interacting* and expecting future events implies some form of environment model and cognitive goings on in the agent. Timestamped propositions are of no great interest to reactive agents.

belief revision and belief update handle inconsistent beliefs differently[8]. An agent using a belief revision system may recover from a state where it holds inconsistent beliefs whereas an agent using a belief update system may continue to hold inconsistent beliefs. Cognitive agents can function whilst holding Inconsistent beliefs though this may carry a performance penalty. Reactive agents are, however ill equipped to deal with such operating states and holding inconsistent beliefs is potentially disastrous.

Belief revision places small restrictions on an agent's prior beliefs, it requires that they form a total preorder which fits with the notion of reactive plans. It would appear that belief revision is a technique well suited to managing beliefs for reactive agents.

Over time a system develops a simple plan for manufacturing a chemical product. Agents provide a reaction vessel with raw materials which spontaneously react to make the desired product. Agents believe that if chemicals X and Y are combined then Z is the result. One day this does not happen. The simple agents are faced with a new observation contradicting a current belief so their belief revision system discards the old beliefs and assumes the new belief that X and Y do not spontaneously react. In a later encounter with a coaching agent the coach sees these discarded beliefs and recognises that this is a reaction that *no longer* occurs. The coach then works with what the agents observer and attempts, using belief revision in combination with other techniques, to identify changes in the environment that may have contributed to this failed action. the coach will have access to data from other agents which may have different percepts and have sensed, perhaps, that the ambient temperature had recently dropped. The coach could conjecture that the temperature had something to do with the change as the ambient temperature was higher in all previous successful cases.

The general agent's *belief fault* is used by by the coach as a trigger for examining its knowledge in an attempt to discover what caused the fault.

3.4 Agent communication

A characteristic of reactive agents is that they do not use a model of their environment. This leads to a requirement for the local environment to provide sufficient information for agents to carry out their task properly[2]. Rather than do this by providing agents with a rich set of communications facilities it was decided to have them leave data directly in the environment. This partially avoids problems with agents using different "names" for objects (mentioned already in section3.2) and sidesteps potential problems with communications channels becoming saturated. Data pertaining to a particular region of an agent's world will be concentrated in and around that region. Data will be placed in the environment by agents leaving internal state records as "patches" in their environment, this is similar to patches in StarLogo (see [16]). This method means that general agents will work, for the most part, with immediately local data but will also have access to smaller amounts of data from elsewhere.

Communications between general agents and coaching agents will be by a dedicated interface allowing the coach access to an agent's history data and the reactive core. The is part of the agent's cognitive module and this "coach interface" will allow agents, either singly or multiply, to communicate directly with dedicated coach agents. The coach will use its greater resources to analyse the collective event histories and may be able to provide improved plans or sequences of actions. Is so then the coach agent will distribute these plans to the connected agents for incorporation into their reactive cores. The coach will also be able to produce state patches incorporating these new plans so that other agents in the field can have access to them.

General agents will be unable to communicate directly with each other so in cases where co-operation between agents is required a triggering mechanism will generate a plausibility for the other agent co-operating. If an agent deems this plausibility sufficient then it will unilaterally start the activity requiring co-operation.

3.5 Analysis of events

Agent cognitive modules will use event histories in a number of ways. Both general and coaching agents will use a backwards looking possible worlds representation of event histories (coach agents will be able to collect and pool this data from several agents) and will apply a variant of standard deontic logic to this in an attempt to identify agent roles by assigning responsibilities to agents to ensure that they *see to it* that certain things are brought about. Horty[17] has presented an account of deontic logic in the context of agency. agency in this account is represented by Chellas's *stit* semantics representing agent's abilities to *see to it* that something is brought about. Coaching agents possess greater resources and will be able to apply a number of techniques, such as data mining, to find relationships between events and agent actions and to identify rules governing the world's behaviour. This will allow the agents to refine existing plans and identify unnecessary actions in a sequence leading to a desired world state.

3.6 A hybrid agent architecture

The planned agent architecture is illustrated in figure 1 and its main components briefly described below.

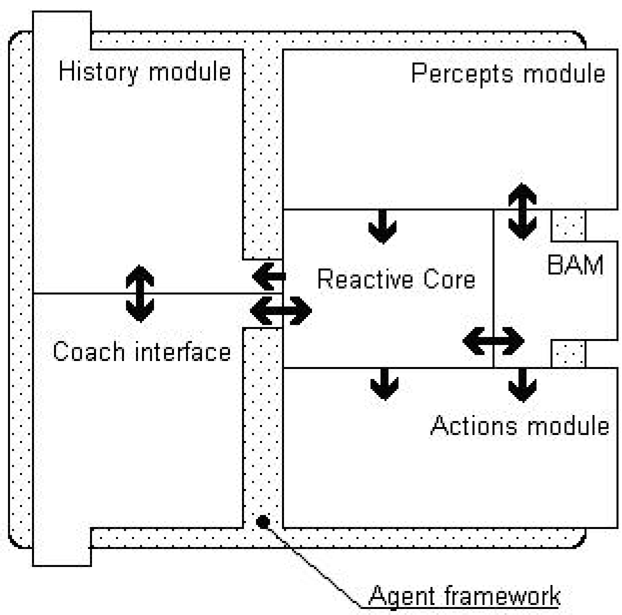

Fig. 1. Planned hybrid agent architecture.

– BAM Basic Agent Module : This is responsible for very basic agent functions moving in the environment, recognising and dealing with state patches.
– Agent framework : This module carries out background tasks and manages communications between the agent's internal modules. The main function of the framework is resource management, it is configured by the BAM and will provide resources to agent modules as instructed.
– Reactive core : This module is responsible for the agents general operation and how it acts in response to environmental data.
– Percepts module : This contains sensors for a subset of world elements and can furnish other modules with environment data.
– Actions Module : The actions module provides an interface which allows the agent to manipulate its environment.
– History module : This module provides general agents with a small store used for maintaining a history of recent events. Coaching agents will have a much larger history module providing storage for a database of behaviours and sample histories. This will allow the coach to acquire sufficient data to allow analysis and refinement of behaviours.
– Coach interface : This is the cognitive part of the hybrid architecture, in non coaching agents this module has two functions, it provides low level assistance when the reactive core has difficulty in making a decision and it manages transactions with coaching agents updating the reactive core as necessary. Dedicated coaching agents are biased more towards cognitive operation and, consequently, the coach interface module plays a much larger role in their operation. The cognitive module will, in such cases, be able to control the reactive module directly.

This hybrid architecture has two main components, the cognitive "coach interface" and the reactive "core". additional modules, with the exception of the BAM, are implementation specific support modules and do not have any control over the agent's operation. The BAM is a minimal reactive core and is part of the agent to ensure that it can at least move and collect state patches when it is newly instantiated in an environment.

4 Related work

Barbeau et al. address similar issues, safety, liveness and agent planning, in [18]. This work differs in a number of respects. Most notably in the use of a global process view and the requirement that the available controllable agent actions and uncontrollable environmental factors are known beforehand. Our method assumes no prior knowledge and has system goals defined in terms of Barbeau's work is similar in that it searches for a solution, our work also searches but the search is carried out by agents and involves discovering and combining atomic actions rather than searching a known set of actions.

5 Conclusion and further work

The novel combination of hybrid agents, simple reactive/cognitive agents using belief revision and more capable cognitive/reactive agents using belief update, can be used to drive system learning. When a simple agent encounters something that leads to an inconsistent belief set it revises its beliefs and moves the *faulted* beliefs to a discard set. Coaching agents will use this discard set and the agent's knowledge in conjunction with pooled discards and knowledge from other agents in an attempt to discover reasons for previously held beliefs becoming faulty. Systems will be specified by describing their safety and liveness properties. A stochastic element in agent behaviour will help to drive discovery and this will be constrained by system safety properties.

It is anticipated that this learning system will allow reactive systems to "evolve" towards a stable system satisfying a required behaviour. The method used should provide some form of reasoning or description of the components of the final behaviour. The ability to handle *faulted* beliefs should give the system the ability to handle disturbances and unusual occurrences. This approach may be useful for generating agent based control systems to replace existing systems that are based on a known specification but have been extensively modified so that the current system configuration is not not thoroughly understood. examples of this may be plant or chemical process controllers with certain products required at certain locations.

References

1. Chang, E.Y., Manna, Z., Pnueli, A. In: Characterization of Temporal Property Classes. Volume 632 of LNCS. Springer-Verlag (1992) 474–486
2. Jennings, N., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems **1** (1998) 7–38 June 2002 Printed.
3. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a bdi-architecture. In Fikes, A.J., Sandewall, E., eds.: proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, KR91, Morgan Kaufmann (1991)
4. Bratman, M.E., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. In Cummins, R., Pollock, J.L., eds.: Philosophy and AI: Essays at the Interface. The MIT Press, Cambridge, Massachusetts (1991) 1–22
5. Georgeff, M., Ingrand, F.: Decision making in an embedded reasoning system. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI (1989)
6. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review **10** (1995) 115–152
7. Cohen, P., Levesque, H.: Persistence, intentions and commitment. In Cohen, P., J., M., Pollack, M., eds.: Intentions in Communication. MIT Press (1990) 33–69
8. Friedman, N., Halpern, J.Y.: Modeling belief in dynamic systems, part II: Revision and update. Journal of Artificial Intelligence Research **10** (1999) 117–167
9. Friedman, N., Halpern, J.Y.: Modeling belief in dynamic systems. part I: Foundations. Artificial Intelligence **95** (1997) 257–316
10. Friedman, N., Halpern, J.: Plausibility measures: A user's guide. In: Uncertainty in AI, Proceedings of the Eleventh Conference. (1995)
11. Wooldridge, M.: An introduction to Multiagent Systems. Wiley (2002)
12. Debenham, J.: A multi-agent architecture for process management accommodates unexpected performance. In: ACM symposium on Applied computing, Como (2000)
13. Strippgen, S., Peters, K.: The other way round! - collaborative communication with agents. In: ACM Agents99. (1999) 108–115
14. Oliveira, E.: Robots as responsible agents. In: IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. Volume 3. (1997) 2275–2279
15. Riley, P., Veloso, M.: Recognizing probabilistic opponent movement models. In Birk, A., Coradeschi, S., Tadokoro, S., eds.: RoboCup-2001: The Fifth RoboCup Competitions and Conferences. Springer Verlag, Berlin (2002) (extended abstract).
16. Resnick, M.: Turtles, termites and traffic jams. MIT Press (1994)
17. Horty, J.: Agency and deontic logic. OUP (2001)
18. Barbeau, M., Kabanza, F., St-Denis, R.: Synthesizing plant controllers using real-time goals. In: Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann (1995) 791–798