On Supporting Rational Behaviour in Real-Time Multi-agent Domains

Ferguson, Innes

On supporting rational behavior in real-time multi-agent domains

Innes A. Ferguson

Interactive Information Group
Institute for Information Technology
National Research Council
Ottawa, ON, Canada K1A 0R6
innes@ai.iit.nrc.ca

Abstract

This paper presents a number of behavioral and architectural design components suitable for controlling rational, autonomous, resource-bounded agents in a class of real-time multi-agent domains. A number of these components have been integrated into the TouringMachine architecture which is described in some detail. An account is also given of a TouringMachine agent's model of practical rationality which is centered around its capability for reasoning abductively about other agents' mental states.

Introduction

This paper is concerned with aspects of the design and implementation of an integrated agent control architecture, the *TouringMachine* architecture (Ferguson 1992; Ferguson 1995), suitable for controlling and coordinating the actions of an autonomous rational agent embedded in a partially-structured, dynamic, multi-agent world.

Implemented as a number of concurrently-operating, latency-bounded, task-achieving control layers, the TouringMachine architecture is able to produce a number of reactive, goal-directed, reflective, and predictive behaviors — as and when dictated by the agent's internal state and environmental context. In particular, TouringMachines (see Figure 1) comprise three such independently motivated layers: a *reactive* layer $\mathcal{R}$ for providing the agent with fast, reactive capabilities for coping with events its higher layers have not previously planned for or modelled (a typical event in a multi-agent road navigation domain, for example, would be the sudden appearance of some hitherto unseen agent or obstacle); a *planning* layer $\mathcal{P}$ for generating, executing, and dynamically repairing hierarchical partial plans (which are used by the agent, for example, when constructing navigational routes to some target destination); and a reflective-predictive or *modelling* layer $\mathcal{M}$ for constructing Belief-Desire-Intention (BDI) models of world entities, including the agent itself, which can be used as a platform for abductively explaining observed behaviors and making predictions about possible future behaviors (Ferguson 1995).

Each control layer is designed to model the agent's world at a different level of abstraction and each is endowed with different task-oriented capabilities. Also, because each layer directly connects world perception to action and can independently decide if it should or should not act in a given state, frequently one layer's proposed actions will conflict with those of another; in other words, each layer is an approximate machine and thus its abstracted world model is necessarily incomplete. As a result, layers are mediated by an enveloping control framework so that the agent, as a single whole, may behave appropriately in each different world situation. Implemented as a combination of inter-layer message passing and context-activated, domain-specific control rules (see Figure 1), the control framework's mediation enables each layer to examine data from other layers, inject new data into them, or even remove data from the layers. This has the effect of altering, when required, the normal flow of data in the affected layer(s). So, in the road driving domain of the *TouringWorld* (Ferguson 1992) for example, the reactive rule in layer $\mathcal{R}$ to prevent an agent from straying over lane markings can, with the appropriate control rule present, be overridden should the agent embark on a plan to overtake the agent in front of it.

Inputs to and outputs from layers are generated in a synchronous fashion, with the context-activated control rules being applied to these inputs and outputs at each synchronization point. The rules, thus, act as filters between the agent's sensors and its internal layers (*suppressors*), and between its layers and its action effectors (*censors*) — in a manner very similar to Minsky's suppressor- and censor-agents (Minsky 1986). Both types of rules are of the *if-then* condition-action type. In the case of censor rules, the conditional parts are conjunctions of statements that test for the presence of particular sensory objects recently stored in the agent's Perception Subsystem. Censor rules' action parts consist of operations to prevent particular sensory objects from being fed as input to *selected* control layers. In the case of suppressor control rules, conditional parts are conjunctions of statements which, besides testing for the presence of particular outgoing action commands in the agent's Action Subsystem, can also test the truth values of various items of the agent's current internal state — in

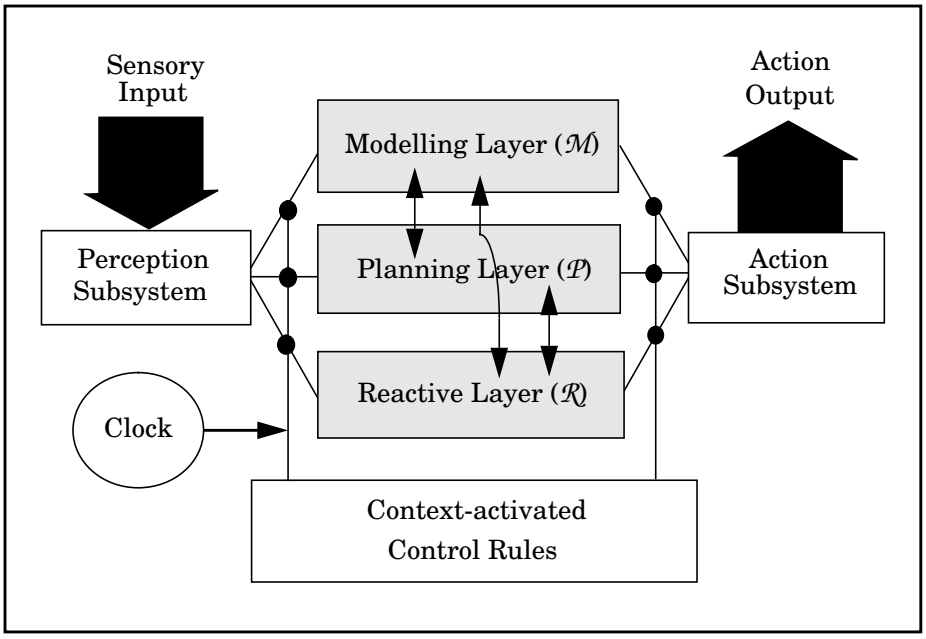

Figure 1. A TouringMachine's mediating control framework.

particular, its current beliefs, desires, and intentions. Suppressor rules' action parts consist of operations to prevent particular action commands from being fed through to the agent's effectors.

Mediation remains active at all times and is largely "transparent" to the layers: each layer acts as if it alone were controlling the agent, remaining largely unaware of any "interference" — either by other layers or by the rules of the control framework — with its own inputs and outputs. The overall control framework thus embodies a real-time opportunistic scheduling regime which, while striving to service the agent's high-level tasks (e.g. planning, causal modelling, counterfactual reasoning) is sensitive also to its low-level, high-priority behaviors such as avoiding collisions with other agents or obstacles.

Modelling Rational Agent Behavior

Like most real-world domains, a TouringMachine's world is populated by multiple autonomous entities and so will often involve dynamic processes which are beyond the control of any one particular agent. For a planner — and, more generally, for a rational agent — to be useful in such domains, a number of special skills are likely to be required. Among these are the ability to monitor the execution of one's own actions, the ability to reason about actions that are outside one's own sphere of control, the ability to deal with actions which might (negatively) "interfere" with one another or with one's own goals, and the ability to form contingency plans to overcome such interference. Georgeff (Georgeff, 1990) argues further that one will require an agent to be capable of coordinating plans of action and of reasoning about the mental state — the beliefs, desires, and intentions — of other entities in the world; where knowledge of other entities' *motivations* is limited or where communication among entities is in some way restricted, an agent will often have to be able to infer such mental state from its observations of entity behavior.

The potential gain from incorporating knowledge level mental modelling capabilities in an autonomous rational agent is that by making successful predictions about entities' activities the agent should be able to detect potential goal conflicts earlier on — thus enhancing the agent's ability to coordinate its actions with other agents (Chandrasekaran, 1994). This would then enable it to make changes to its own plans in a more effective manner than if it were to wait for these conflicts to materialize. Goal conflicts can occur within the agent itself (for example, the agent's projected time of arrival at its destination exceeds its original deadline or the agent's layer $\mathcal{R}$ effects an action which alters the agent's trajectory) or in relation to another agent (for example, the agent's trajectory intersects that of another agent). Associated with the different goal conflicts that are known to the agent are a set of conflict resolution strategies which, once adopted, typically result in the agent taking some action or adopting some new intention.

The structures used by an agent to model an entity's behavior are time indexed 4-tuples of the form $\langle C, B, D, I \rangle$, where C is the entity's *Configuration*, namely *(x,y)*-location, speed, acceleration, orientation, and signalled communications; B is the set of *Beliefs* ascribed to the

entity; *D* is its ascribed list of prioritized goals or *Desires*; and *I* is its ascribed plan or *Intention* structure. Plan ascription or recognition has been realized in TouringMachines as a process of *scientific theory formation* which employs an abductive reasoning methodology similar to that of the Theorist default/diagnostic reasoning system (Poole *et al*., 1986) — more on this shortly.

These Belief-Desire-Intention (BDI) models used by an agent are, in fact, filled-in templates which the agent obtains from an internal model library. While all templates have the same basic 4-way structure, they can be made to differ in such aspects as the depth of information that can be represented or reasoned about (for example, a particular template's *B* component might dictate that modelled beliefs are to be treated as defeasible), initial default values provided, and computational resource cost. The last of these will subsequently be taken into account each time the agent makes an inference from the chosen model.

Reasoning from a model of an entity essentially involves looking for the interaction of observation and prediction; that is, for any discrepancies between the agent's *actual* behavior and that *predicted* by its model or, in the case of a self-model, between the agent's actual behavior and that *desired* by the agent. Model-based reasoning in TouringMachines specifically comprises two phases: *explanation* and *prediction*. During the explanation phase, the agent attempts to generate plausible or inferred explanations about any entity (object/agent) behaviors which have recently been observed. Explanations (models) are then used in detecting discrepancies between these entities' current behaviors and those which had been anticipated from previous encounters. If any such behavioral discrepancies are detected, the agent will then strive to infer, via intention ascription, plausible explanations for their occurrence.

Theory Formation and Selection using Theorist

Theorist (Poole *et al*., 1986) is a logic programming system for constructing scientific theories — that is, for constructing explanations of *observations* in terms of various *facts* and *hypotheses*. Theorist is a system for both representation and reasoning. A Theorist knowledge base consists of a collection of first order clausal form logic formulae which can be classified as: *(i)* a closed set of consistent formulae or facts, *F*, which are known to be true in the world; *(ii)* the possible hypotheses, Δ, which can be accepted as part of an explanation; and *(iii)* the set of observations, *G*, which have to be explained. Given these, the Theorist reasoning strategy attempts to accumulate consistent sets of facts and instances of hypotheses as explanations for which the observations are logical consequences. An explanation or *theory* is then a subset of the possible hypotheses which are consistent and which imply the observations. More formally, *G* is said to be *explainable* if there is some subset *D* of Δ such that

$$F \cup D \models G \text{ and}$$

$$F \cup D \text{ is consistent.}$$

D is said to be a *theory that explains G*. *D* should then be seen as a "scientific theory" (Poole et al., 1986, page 4).

Theorist has been described as both a theory and an implementation for default and *abductive* reasoning (Poole, 1988). One of the several ways in which Theorist can been used, then, is for performing *abductive diagnosis*; namely, finding a set of causes (for example, diseases) which can imply the observed effects (for example, patients' symptoms). Now, by taking the system or artifact that is being diagnosed as the entity that our TouringMachine agent is modelling, and by re-interpreting "symptoms" as the entity's observed actions, then the causes behind this entity's actions can be regarded as the entity's intentions.[1] Note, then, that in the context of TouringMachines, the process of finding the intentions which are the cause of some other entity's actions is effectively one of performing plan inference or recognition (Carberry, 1990).

Theorist is invoked once for every one of the agent's entity models that displays a model-entity (or expectation-observation) discrepancy. In particular, Theorist is called by supplying it with the name of the agent that is doing the modelling, the name of the entity that is being modelled, the agent's observations of that entity (that is, all relevant details of the entity's current configuration as modelled by the agent), and the current value of the agent's internal clock. Theorist's reasoning strategy then tries to accumulate consistent sets of facts and instances of hypotheses, or defaults, as explanations for which the observations are logical consequences. The facts and defaults reside in the *Theorist KBase*, a knowledge base containing a domain model of the TouringWorld expressed in terms of the various "faults" that can be used to explain entities' "errant" behaviors. In the present context, faults can be viewed as the causes for — or the intentions behind — why certain events — or certain observed actions of some entity — might have occurred in the world. Details on agents' domain models can be found elsewhere (Ferguson, 1992).

Generating Expectations and Closing the Loop

Once all BDI model discrepancies have been identified and their causes inferred, predictions are formed by temporally projecting those parameters that make up the modelled entity's configuration vector *C* in the context of the current world situation and the entity's ascribed intention. The space-time projections (in effect,

[1]It should be noted that, for the purpose of simplifying the implementation of TouringMachines, agents' beliefs and desires are assumed to be common and so can be ignored during the theory formation process.

knowledge-level simulations) thus created are used by the agent to detect any potential interference or goal conflicts among the modelled entities' anticipated/desired actions. Should any conflicts — intra- or inter-agent — be identified, the agent will then have to determine how such conflicts might best be resolved, and also which entities will be responsible for carrying out these resolutions. Determining such resolutions, particularly where multiple goal conflicts are involved, will require consideration of a number of issues, including the priorities of the different goals affected, the space-time urgency of each conflict, rights-of-way protocols in operation, as well as any environmental and physical situational constraints (for example, the presence of other entities) or motivational forces (for example, an agent's own internal goals) that may constrain the possible actions that the agent can take (Ferguson, 1992; Ferguson, 1995).

Resource-boundedness and rationality in dynamic worlds

Successful operation in a dynamic domain such as the TouringWorld will require real-time responses to a range of unanticipated events and planning exceptions. Real-time responses might be required, for example, if an agent is to avoid missing an important task deadline or, more importantly, if it is to prevent the catastrophic consequences of colliding with another agent or obstacle. In such environments, a perfectly rational agent would always bring all of the information in its memory and in the environment to bear on its various perception, decision-making, and execution tasks.

Ideally one would wish for an agent control architecture which, on the one hand, was capable of generating rational, minimal, and provably correct action sequences, and on the other, was capable of flexibly and robustly dealing with the real-time pressures that are characteristic of dynamic multi-agent domains. In toy domains like the blocks worlds of classical AI planning fame (Fikes & Nilsson 1990), it may well be possible for a single agent to guarantee the generation and execution of optimal action sequences. In realistic domains, however, the requirements to behave at once correctly, flexibly, *and* robustly conflict with each other, agents often having to rely on heuristic or satisficing methods of decision-making to ensure the successful completion of their tasks. As Simon puts it: the agent will have a choice between "optimal decisions for an imaginary simplified world or decisions that are "good enough," that satisfice, for a world approximating the complex real one more closely." (Simon 1981, page 35).

The TouringMachine architecture is aimed at supporting the development of real-time embedded agents. In such agents, there is not always enough time — nor indeed computational power — to make use of all available knowledge for each and every task-related activity. To cope with such constraints, TouringMachines are designed to make use of latency-bounded heuristic functions to simplify some of their decision-making processes. For example, TouringMachines satisfice with respect to planning by employing a limited search that is directed by stored heuristic knowledge. Also, TouringMachines' layers $\mathcal{P}$ (planning) and $\mathcal{M}$ (modelling) are designed so as not to process all of the sensory input collected by their Perception Subsystems; instead they make use of a selective attention mechanism which helps them to focus only on those environmental features which are deemed relevant to the agent's situational and task-related needs. A consequence of this, however, is that, at times (for instance, in sufficiently time-constrained situations), TouringMachines may well generate plans which are less than optimal, some even failing to achieve their initially intended tasks unless appropriately repaired. (Increasingly, it should be noted, researchers are accepting — and so explicitly dealing with the issue — that realistic physical systems will inevitably make some such "mistakes" (Brooks 1986; Russell & Wefald 1991).)

Another complication which arises vis-à-vis guaranteeing rational behavior is the fact that TouringMachines have multiple goals, some of which, on occasion, will conflict with one another. For example, the act of slowing down to avoid colliding with another agent will likely conflict with the agent's main time-constrained navigational task, especially if this task's deadline was reasonably tight to begin with. Similarly, if an agent waiting at a red light determines that it is about to be hit from behind by another entity, a TouringMachine will, by design, reactively move to avoid the collision — even if this results in driving through the red light.[2] So, because a TouringMachine will not be able to attend to all of its goals simultaneously, some of its actions may occasionally be inconsistent with one or more of these goals. Note, however, that such actions typically serve a protective purpose for the agent and so should not be regarded as undesirable. On the contrary, since in dynamic domains agent robustness and survival are usually considered more important than correctness, protective actions — perfectly rational or otherwise — must surely be considered indispensable.

According to Bratman *et al.* (Bratman, Israel & Pollack 1988), a rational agent is one which is *committed* to doing what it plans; and as such, only under exceptional circumstances — for example, when an explicit impediment is detected or when some task delay can be foreseen — should the agent undertake to alter its plans. Knowing when to reconsider committed plans is not simple: as Minsky (Minsky 1986, page 163) points out, "Too much commitment leads to doing only one single thing; too little concern produces aimless wandering." What this suggests,

[2] As explained elsewhere (Ferguson 1992), a TouringMachine builds and executes run-time plans to stop at red lights in order to satisfy `obey-regulations` — one of several layer $\mathcal{M}$ goals. In this example, then, the action of going through a red light will conflict with the agent's goal `obey-regulations`.

in fact, is the existence of a "tension" between the *stability* that an agent's plans must have in order to provide a focus for the agent's deliberative reasoning processes, and the *revocability* that the same plans must also exhibit, given that they will only ever have been conceived with partial information about the agent's past, present, and future states.

Long-term rational behavior, then, would appear to result from continually balancing the tension between plan stability (goal-orientedness, future-directedness) on the one hand, and plan revocability (reactivity, flexibility) on the other. As Maes (Maes 1990) suggests, obtaining the right balance of such behavioral characteristics will depend on particular aspects of the agent's task and environment: for example, the precision with which the task must be carried out, the time available for making control decisions, or the degree of predictability in the agent's surrounding environment. In fact, as shown in a number of different experiments (Ferguson 1992; Ferguson 1994) — and confirmed by several other investigations into resource-bounded agency (Bratman, Israel & Pollack 1988, Cohen *et al.* 1989; Pollack & Ringuette 1990) — the production of rational behavior can also be seen to depend on characteristics of the agent's own internal design or configuration — for example, the agent's degree of sensitivity to unanticipated environmental change.

Because TouringMachines will almost constantly be in a state of perceptual, cognitive, and action overload, they will generally not be able to perform all potential operations in a timely manner. This, it should be noted, has less to do with precisely how fast TouringMachines can operate than with the fact that they are inherently rationally bounded. Following Hayes-Roth's philosophy, the aim in designing TouringMachines, therefore, is not to create agents which are optimized for the performance of a single pre-determined task; rather, it is to design a control architecture which is capable of satisficing performance over a *range* of tasks, each potentially differing in terms of their required functionality, knowledge sources, and associated time constraints (Hayes-Roth 1990). In the TouringMachine framework, then, achieving rational behavior should really be viewed as *one of many* of the agent's objectives (guaranteed real-time performance would be another, for example), which it will be able to achieve to a greater or lesser extent depending on prevailing environmental conditions and the availability of necessary resources.

References

Bratman, M.E.; Israel, D.J.; and Pollack, M.E. 1988. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):349—355.

Brooks, R.A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14-23.

Carberry, S. 1990. *Plan Recognition in Natural Language Dialogue*, MIT Press: Cambridge, MA.

Chandrasekaran, B. 1994. Understanding control at the knowledge level. In *Working Notes AAAI Fall Symposium on Control of the Physical World by Intelligent Agents,* New Orleans, LA, pp. 19-26.

Cohen, P.R.; Greenberg, M.L.; Hart, D.M.; and Howe, A.E. 1989. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32-48.

Ferguson, I.A. 1992. *TouringMachines: An architecture for dynamic, rational, mobile agents*. Ph.D. thesis, Computer Laboratory, University of Cambridge, Cambridge, UK.

Ferguson, I.A. 1995. On the role of BDI modeling for integrated control and coordinated behavior in autonomous agents. *Applied Artificial Intelligence*, 9(4). In press.

Fikes, R.E., and Nilsson, N.J. 1990. STRIPS: A new approach to the application of theorem proving to problem solving. In J. Allen, J. Hendler, and A. Tate, eds., *Readings in Planning*, pages 88-97. Morgan Kaufmann: San Mateo, CA.

Georgeff, M.P. 1990. Planning. In J. Allen, J. Hendler, and A. Tate, eds., *Readings in Planning*, pages 5-25. Morgan Kaufmann: San Mateo, CA.

Hayes-Roth, B. 1990. Architectural foundations for real-time performance in intelligent agents. *The Journal of Real-Time Systems*, 2:99-125.

Maes, P. 1990. Situated agents can have goals. *Robotics and Autonomous Systems*, 6(1&2):49-70.

Minsky, M.L. 1986. *The Society of Mind.* Simon and Schuster: New York, NY.

Pollack, M.E., and Ringuette, M. 1990. Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings Conference of the American Association for Artificial Intelligence*, pages 183-189.

Poole, D.L., Goebel, R.G., and Aleliunas, R. 1986. Theorist: A logical reasoning system for defaults and diagnosis. Research Report CS-86-06, University of Waterloo, Waterloo, ON, February.

Poole, D. 1988. Representing knowledge for logic-based diagnosis. In *Proceedings International Conference on Fifth Generation Computer Systems*, pages 1282—1289.

Russell, S., and Wefald, E. 1991. *Do the Right Thing — Studies in Limited Rationality*. MIT Press: Cambridge, MA.

Simon, H.A. 1981. *The Sciences of the Artificial.* MIT Press: Cambridge, MA, 2nd edition.