



NRC Publications Archive Archives des publications du CNRC

Autonomous Agent Control: A Case for Integrating Models and Behaviors

Ferguson, I.A.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=029c6cef-616f-4575-95f7-1d96335fb22c>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=029c6cef-616f-4575-95f7-1d96335fb22c>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



Autonomous Agent Control: a Case for Integrating Models and Behaviors*

Innes A. Ferguson

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

Abstract

It is becoming widely accepted that neither purely reactive nor purely deliberative control techniques are capable of producing the range of behaviors required of intelligent computational agents in dynamic, unpredictable, multi-agent worlds. This paper presents a new architecture for controlling autonomous agents, building on previous work addressing reactive and deliberative control methods. The proposed multi-layered architecture allows a resource-bounded, goal-directed agent to reason predictively about potential conflicts by constructing causal theories or models which explain other agents' observed behaviors and hypothesize their goals and intentions; at the same time it enables the agent to operate autonomously and to react promptly to changes in its real-time environment.

A principal aim of this research is to understand the role different functional capabilities play in constraining an agent's behavior under varying environmental conditions. To this end, an experimental testbed has been constructed comprising a simulated multi-agent world in which a variety of agent configurations and behaviors have been investigated. A number of experimental findings are reported.

1 Introduction

The computer-controlled operating environments at such facilities as automated factories, nuclear power plants, telecommunications installations, and information processing centers are continually becoming more complex. As this complexity grows, it will be increasingly difficult to control such environments with centralized

management and scheduling policies that are both robust in the face of unexpected events and flexible at dealing with operational and environmental changes that might occur over time. One solution to this problem which has growing appeal is to distribute, along such dimensions as space and function, the control of such operations to a number of intelligent, task-achieving robotic or computational agents.

Most of today's computational agents are limited to performing a relatively small range of well-defined, pre-programmed, or human-assisted tasks. Operating in real world domains means having to deal with unexpected events at several levels of granularity — both in time and space, most likely in the presence of other independent agents. In such domains agents will typically perform a number of complex simultaneous tasks requiring some degree of attention to be paid to environmental change, temporal constraints, computational resource bounds, and the impact agents' shorter term actions might have on their own or other agents' longer term goals. Also, because agents are likely to have incomplete knowledge about the world and will compete for limited and shared resources, it is inevitable that, over time, some of their goals will conflict. Any attempt to construct a complex, large-scale system in which all envisaged conflicts are foreseen and catered for in advance is likely to be too expensive, too complex, or perhaps even impossible to undertake given the effort and uncertainty that would be involved in accounting for all of one's possible future equipment, design, management, and operational changes.

* This research was conducted while the author was a doctoral candidate at the Computer Laboratory, University of Cambridge, Cambridge, UK.

Now, while intelligent agents must undoubtedly remain reactive in order to survive, some amount of strategic or predictive decision-making will also be required if agents are to handle complex goals while keeping their long-term options open. On the other hand, agents cannot be expected to model their surroundings in every detail as there will simply be too many events to consider, a large number of which will be of little or no relevance anyway. Not surprisingly, it is becoming widely accepted that neither purely reactive [Bro86, AC87, Sch87] nor purely deliberative [DM90, Sho90, VB90] control techniques are capable of producing the range of robust, flexible behaviors desired of future intelligent agents. What is required, in effect, is an architecture that can cope with uncertainty, react to unforeseen incidents, and recover dynamically from poor decisions. All of this, of course, on top of accomplishing whatever tasks it was originally assigned to do.

This paper is concerned with the design and implementation of a novel integrated agent control architecture, the *TouringMachine* architecture [Fer91, Fer92a, Fer92b, Fer92c], suitable for controlling and coordinating the actions of autonomous rational agents embedded in a partially-structured, dynamic, multi-agent world. Upon carrying out an analysis of the intended *TouringMachine* task domain — that is, upon characterizing those aspects of the intended real-time indoor navigation domain that would most significantly constrain the *TouringMachine* agent design — and after due consideration of the requirements for producing autonomous, effective, robust, and flexible behaviors in such a domain, the *TouringMachine* architecture has been designed through integrating a number of reactive and *suitably designed* deliberative control functions.

2 TouringMachines

Implemented as a number of concurrently-operating, latency-bounded, task-achieving control layers, the resulting *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, 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 behavioral device models of world entities, including the agent itself, which can be used as a platform for explaining observed behaviors and making predictions about possible future behaviors (more on this below).

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 2), 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. (The term *data* here covers sensed input to and action output from layers, the contents of inter-layer messages, as well as certain rules or plans residing within layers.) This has the effect of

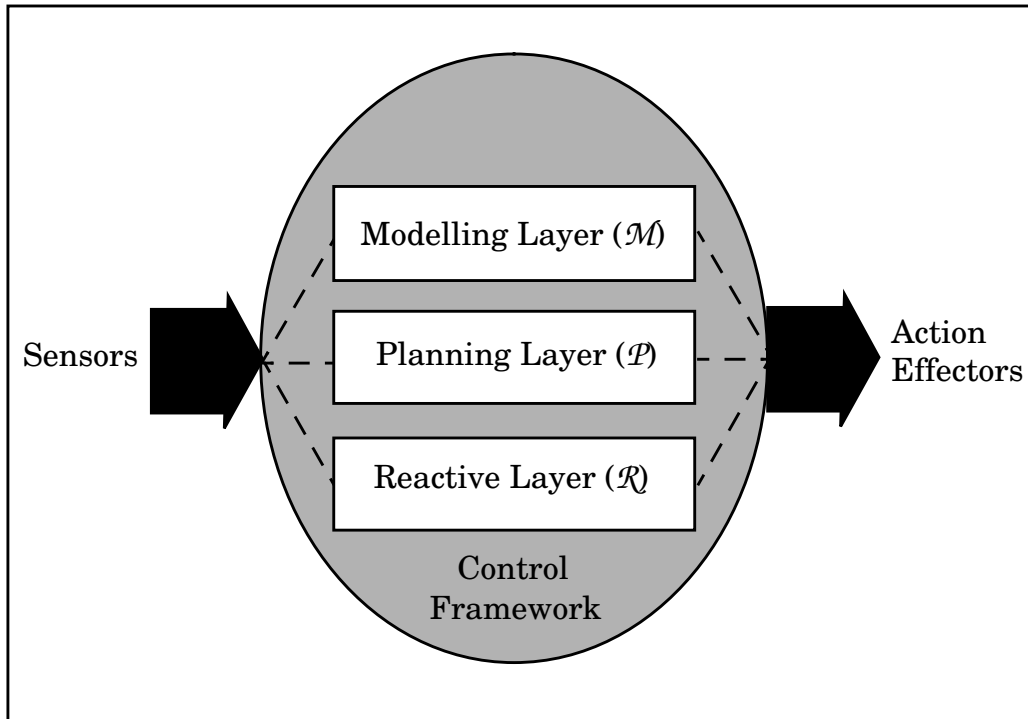


Figure 1: The TouringMachine Architecture.

altering, when required, the normal flow of data in the affected layer(s). So, in a road driving domain 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 [Min86]. 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, counter-factual reasoning) is sensitive also to its low-level, high-priority behaviors such as avoiding collisions with other agents or obstacles.

3 Modelling Agent Behavior

Like most real-world domains, a Touring-Machine’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 an 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

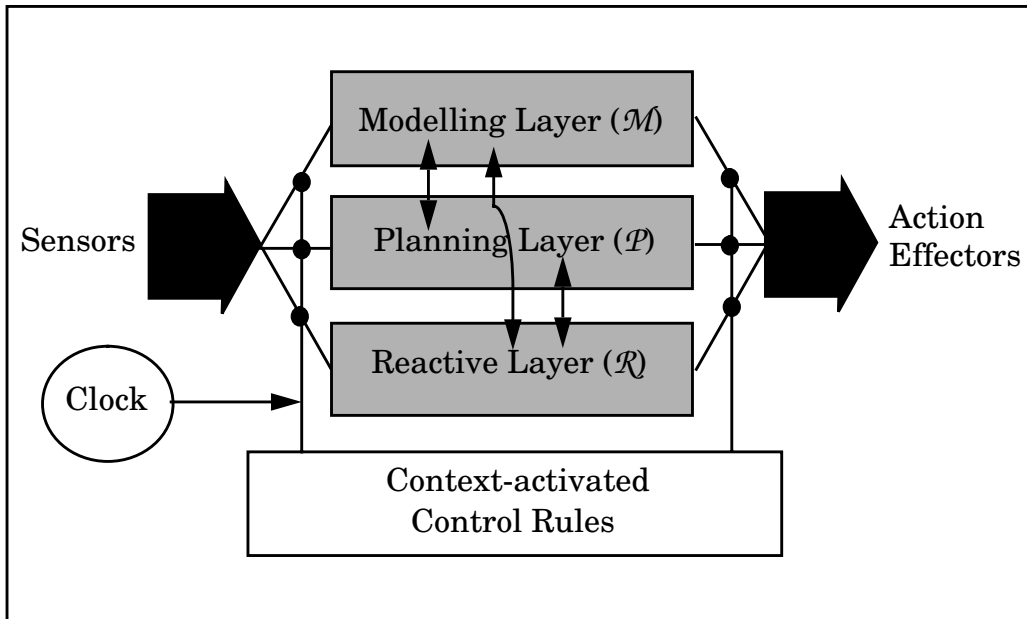


Figure 2: A TouringMachine's mediating control framework.

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 [Geo90] 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, goals, 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. Kirsh, in addition, argues that for survival in real-world, human style environments, agents will require the ability to frame and test hypotheses about the future and about other agents’ behaviors [Kir91].

The potential gain from incorporating causal device or mental modelling capabilities in an autonomous agent is that by making successful predictions about entities’ activities the agent should be able to detect potential goal conflicts earlier on. 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 [PGA86].

The device 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” [DH88]; 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.

Once all 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, 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 (e.g. the presence of other entities) or motivational forces (e.g. an agent's own internal goals) that may constrain the possible actions that the agent can take [Fer92c].

4 Experimenting with Touring-Machines

The research presented here adopts a fairly pragmatic approach toward understanding how complex environments might constrain the design of agents, and, conversely, how different task constraints and functional capabilities within agents might combine to produce different behaviors. In order to evaluate TouringMachines, a highly instrumented, parametrized, multi-agent simulation testbed has been implemented in conjunction with the TouringMachine control architecture. The testbed provides the user with a 2-dimensional world — the *TouringWorld* — which is occupied by, among other things, multiple TouringMachines, obstacles, walls, paths, and assorted information signs. World dynamics are realized by a discrete event simulator which incorporates a plausible world updater for enforcing “realistic” notions of time and motion, and which creates the illusion of concurrent world activity through appropriate action scheduling. Other processes handled by the simulator include a facility for tracing agent and environmental parameters, a statistics gathering package for agent performance analysis, a mechanism enabling the testbed user to control the motion of a chosen agent, and several text and

graphics windows for displaying output. By enabling the user to specify, visualize, measure, and analyze any number of user-customized agents in a variety of single- and multi-agent settings, the testbed provides a powerful platform for the empirical study of autonomous agent behavior.

A number of experiments have been carried out on TouringMachines which illustrate, in particular, that the balance between goal-orientedness (effectiveness) and reactivity (robustness) in agents can be affected by a number of factors including, among other things, the level of detail involved in the predictions agents make about each other, the degree of sensitivity they demonstrate toward unexpected events, and the proportion of total agent resources that are made available for constructing plans or building mental models of other agents. Other experiments point toward a trade off between the reliability and the efficiency of the predictions an agent can make about the future (this turns out to be an instance of the well known *extended prediction problem* [SM90]). Yet other experiments have been carried out which suggest that predicting future world states through causal modelling of agents' mental states, can, in certain situations, prove useful for promoting effective coordination between agents with conflicting goals. To illustrate some of the diverse opportunities for analysis which are afforded by the TouringMachine testbed, one particular experiment that illustrates the role of causal modelling of agent behavior will now be described in some detail.

4.1 Counterfactual Reasoning: why modelling other agents' intentions can be useful

In constructing and projecting models of other world entities, a TouringMachine must constrain its modelling activities along a number of dimensions. Implemented as user-definable parameters, these layer \mathcal{M} constraints can be used to define such things as the tolerable deviations between the agent's actual and desired headings, the length of time into the future over which the agent's

conflict detection predictions will apply, the rate at which the agent updates its models, and the total number of per-clock-cycle resources available for constructing models. One other layer \mathcal{M} parameter which is of particular interest here is **ConflictResolutionDepth** — the parameter which fixes the number of levels of counterfactual reasoning the agent should undertake when projecting entities' models to discover possible future goal conflicts. In general, when constructing model projections at counterfactual reasoning level N , an agent will take into account any conflicts *plus any actions resulting from the anticipated resolutions to these conflicts* which it had previously detected at level $N-1$. Values of **ConflictResolutionDepth** which are greater than 1, then, give agents the flexibility to take into account — up to some fixed number of nested levels of modelling — any agent's responses to any other agent's responses to any predicted conflicts.

In the scenario of Figure 3, two TouringMachine agents can be seen following independent routes to one destination or another. The interesting agent to focus on here — the one whose configuration is to be varied — is `agent1` (the round one). The upper left-hand frame of Figure 3 simply shows the state of the world at time $T = 15.5$ seconds. Throughout the scenario, each agent continually updates and projects the models they hold of each other, checking to see if any conflicts might be “lurking” in the future. At $T = 17.5$ (upper right-hand frame of Figure 3), `agent1` detects one such conflict: an `obey-regulations` conflict which will occur at $T = 22.0$ between `agent2` (chevron-shaped) and the traffic light (currently red).³ Now, assuming `agent1` is just far enough away from the traffic light so that it does not, within its parametrized conflict detection horizon,

3. All agents possess the homeostatic goal `obey-regulations` which, in this particular example, will trigger a goal conflict if the agent in question (`agent2`) is expected to run through the red traffic light.

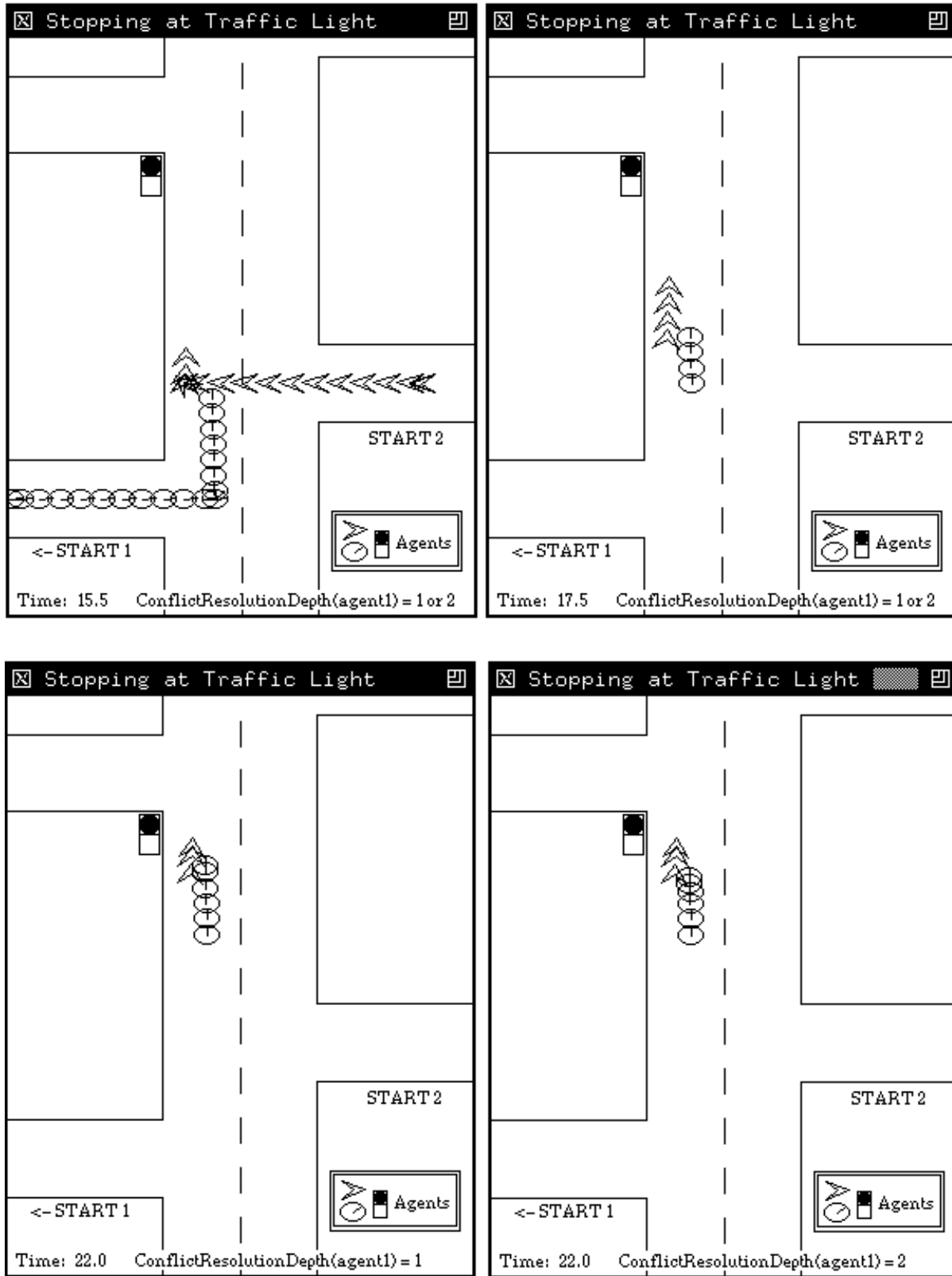


Figure 3: Altering the value of an agent's **ConflictResolutionDepth** parameter can affect the timeliness and effectiveness of any predictions it might make.

see any conflict between itself and the traffic light, then, if `agent1` is configured with **ConflictResolutionDepth** = 1, it will predict the impending conflict between `agent2` and the traffic light, as well as the likely event of `agent2` altering its intention to `stop-at-light` so that it will come to a halt at or around $T = 22.0$. If, on the other hand, `agent1` is configured with **ConflictResolutionDepth** = 2, not only will it predict the same conflict between `agent2` and the traffic light and the resolution to be realized by this entity, but it will also, upon hypothesizing about the world state *after* this conflict resolution is realized, predict another impending conflict, this second one involving itself and the soon to be stationary `agent2`.

The observable effects of this parameter difference are quite remarkable. When `agent1` is configured with **ConflictResolutionDepth** = 1, it will not detect this second conflict — the one between itself and `agent2` — until one clock cycle later; that is, at time $T = 18.0$ instead of at $T = 17.5$. Due to the proximity of the two agents, the relatively high speed of `agent1`, and the inevitable delay associated with any change in intention or momentum, this 0.5 second delay proves to be sufficiently large to make `agent1` realize too late that `agent2` is going to stop; an inevitable rear-end collision therefore occurs at $T = 22.0$ (Figure 3, lower left-hand frame).⁴ Configured with **ConflictResolutionDepth** = 2 (Figure 3, lower right-hand frame), `agent1` ends up having enough time — an extra 0.5 seconds — to adopt and realize the appropriate intention `stop-behind-agent`, thereby avoiding the collision that would otherwise have occurred.

Having the flexibility to reason about the interactions between other world entities (for

4. In fact, this collision need not be “inevitable”: in this scenario both `agent1` and `agent2` have been configured with fairly insensitive (not very robust) layer \mathcal{R} reactions, primarily to emphasize the different behaviours that *could* result from different parametrizations of agents’ modelling capabilities.

example, between `agent2` and the traffic light) and to take into account the likely future intentions of these entities (for example, `stop-at-light`) can enable TouringMachines like `agent1` to make timely and effective predictions about the changes that are taking place or that are likely to take place in the world. In general, however, knowing how deeply agents should model one another is not so clear: since the number of layer \mathcal{M} resources required to model world entities is proportional to both the *number* of entities modelled and the (counterfactual reasoning) *depth* to which they are modelled, agents will ultimately have to strike a balance between breadth of coverage (more entities modelled, little detail) and depth of coverage (less entities, more detail). This issue is investigated in more detail elsewhere [Fer92c].

5 Conclusions

Through the above and a number of other single- and multi-agent coordination experiments addressing such issues as the production of emergent behavioral patterns, the TouringMachine architecture has been shown to be feasible and that, when suitably configured, can endow rational autonomous agents with appropriate levels of effective, robust, and flexible control for successfully carrying out multiple goals while simultaneously dealing with a number of dynamic multi-agent events.

The integration of a number of traditionally expensive deliberative reasoning mechanisms (for example, causal modelling and hierarchical planning) with reactive or behavior-based mechanisms is a challenge which has been addressed in the TouringMachine architecture. Additional challenges such as enabling effective agent operation under real-time constraints and with bounded computational resources have also been addressed. The result is a novel architectural design which can successfully produce a range of useful behaviors required of sophisticated autonomous agents embedded in complex environments.

The research presented here is ongoing;

current work on the TouringMachine agent architecture includes an effort to generalize further the TouringWorld testbed, in particular, by separating the definition of the agent's domain of operation (description of the environment, initial goals to accomplish, criteria for successful completion of goals) from the configuration (capabilities, internal parameters and constraints) of the agent itself. Another aspect of the current work is to identify and incorporate new capabilities in order to extend the behavioral repertoire of agents; capabilities being considered at present include, among others, reinforcement learning, user modelling, episodic memory management, and WWW navigation.

References

- [AC87] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings Conference of the American Association for Artificial Intelligence*, pages 268—272, 1987.
- [Bro86] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14—23, 1986.
- [DH88] Randall Davis and Walter Hamscher. Model-based reasoning: Troubleshooting. In Howard E. Shrobe and AAAI, editors, *Exploring Artificial Intelligence*, pages 297—346. Morgan Kaufmann: San Mateo, CA, 1988.
- [DM90] Edmund H. Durfee and Thomas A. Montgomery. A hierarchical protocol for coordinating multiagent behaviors. In *Proceedings Conference of the American Association for Artificial Intelligence*, pages 86—93, 1990.
- [Fer91] Innes A. Ferguson. Toward an Architecture for Adaptive, Rational, Mobile Agents (extended abstract). In M.P. Georgeff and A.S. Rao (eds.) *Proc. of the IJCAI-91 Workshop on Theoretical and Practical Design of Rational Agents*, Sydney, Australia, August, 1991.
- [Fer92a] Innes A. Ferguson. Touring-Machines: Autonomous Agents with Attitudes. *IEEE Computer*, 25(5), 51—55, 1992.
- [Fer92b] Innes A. Ferguson. Toward an Architecture for Adaptive, Rational, Mobile Agents. In E. Werner and Y. Demazeau (eds.) *Decentralized AI 3*, Elsevier Science (North Holland): Amsterdam, 1992.
- [Fer92c] Innes A. Ferguson. *Touring-Machines: An Architecture for Dynamic, Rational, Mobile Agents*. Ph.D. thesis, Computer Laboratory, University of Cambridge, Cambridge UK, 1992.
- [Geo90] Michael P. Georgeff. Planning. In James Allen, James Hendler, and Austin Tate, editors, *Readings in Planning*, pages 5—25. Morgan Kaufmann: San Mateo, CA, 1990.
- [Kir91] David Kirsh. Today the earwig, tomorrow man? *Artificial Intelligence*, 47:161—184, 1991.
- [Min86] Marvin L. Minsky. *The Society of Mind*. Simon and Schuster: New York, NY, 1986.
- [PGA86] David L. Poole, Randy G. Goebel, and Romas Aleliunas. Theorist: A logical reasoning system for defaults and diagnosis. Research Report CS-86-06, University of Waterloo, Waterloo, Ont., February 1986.
- [Sch87] M.J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings International Joint Conference on Artificial Intelligence*, pages 1039—1046, 1987.
- [Sho90] Yoav Shoham. AGENT0: A simple agent language and its interpreter. In *Proceedings Conference of the American Association for Artificial Intelligence*, pages 704—709, 1991.
- [SM90] Yoav Shoham and Drew McDermott. Problems in formal temporal reasoning. In James Allen, James Hendler, and Austin Tate, editors, *Readings in Planning*, pages 581—587. Morgan Kaufmann: San Mateo, CA, 1990.
- [VB90] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6(1):41—60, 1990.