AI Magazine Volume 23 Number 1 (2002) (© AAAI)

Dynamic Logic
A Review

Andrés Silva

The real world is dynamic, and any intelligent perception of the world should include the concept of time. Remember that time and space are a priori conditions of human perception in Kant's philosophy. On the one hand, time is inherent to action and change; on the other, action and change are possible because of the passage of time. According to McDermott, "Dealing with time correctly would change everything in an AI program" (McDermott 1982, p. 101).

It should not be surprising then that temporal reasoning has always been a very important topic in many fields of AI, particularly areas dealing with change, causality, and action (planning, diagnosis, natural language understanding, and so on). AI developments based on temporal reasoning lead to general theories about time and action, such as McDermott's (1982) temporal logic, Vilain's (1982) theory of time, and Allen's (1984) theory of action and time. Work on the application of these results has taken place in fields such as planning and medical knowledge-based systems.

However, action and change are not an exclusive interest of AI. In mainstream computer science, any execution of a "traditional" computer program is considered to perform an action that leads to a change of state. From this point of view, the field of program verification, traditionally focused on the correctness of actions carried out by program executions, can potentially provide AI with many approaches suitable for dealing with action and change. Temporal logic and dynamic logic are two of the approaches that have been used in the fields of both AI and program verification, temporal logic being the most popular. Both temporal and dynamic logic provide alternative applications of modal logic to program specification and verification. The main difference between the two is that temporal logic is endogenous, and dynamic logic is exogenous. A logic is *exogenous* if programs (actions) are explicit in the language. Temporal logic is *endogenous*, so its programs (actions) are never explicit in the language. Dynamic logic subsumes temporal logic.

Harel, D.; Kozen, D.; and Tiuryn, J. *Dynamic Logic*. Cambridge, Massachusetts, The MIT Press, 2000, 459 pp., $50.00, ISBN 0-262-08289-6.

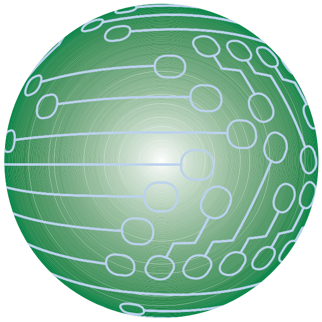
Some cross-fertilization has already take place between AI, temporal logic, and dynamic logic. I focus on dynamic logic, which is the topic that is covered at length in the book under review. *Dynamic logic* is an approach to program verification with strong AI potential. One of the most prominent uses of dynamic logic in AI was Moore's (1990) approach. Moore formalized some issues related to agency, with a focus on what an agent needs to know to be able to perform an action. For more information on this topic and a clear demonstration of the usefulness of dynamic logic for agent reasoning and action, see the survey by Meyer (1999). Also, some research in knowledge engineering inspired by, or making use of, dynamic logic has been published van Harmelen and Balder (1992) and Fensel (1995).

Dynamic logic is an eclectic approach to program verification, as is evidenced by its history. This history starts with the pragmatics of programming, that is, the study of the actions that programs perform and the correctness of these actions. This has been a major issue in computer science since Dijkstra's (1968) attacks on the GOTO statement. Perhaps the most popular formal approach aimed at proving program correctness is Hoare's (1969), which is based on correctness assertions. In Hoare's logic, statements of the form {a}*P*{b} say that if program *P* starts in an input state satisfying *a*, then if and when *P* halts, it does so in a state satisfying *b*. Hoare provided some inference rules used to infer assertions about programs from assertions about other programs.

In 1976, Pratt (1976) made the connection between program logic and *modal logic,* an older tradition in which classical logic is extended with modalities. The two most important modalities used in modal logic are necessity and possibility, whose respective modal operators are $\Box$ and $\Diamond$. Therefore, if *f* is a formula, then so are $\Box f$ and $\Diamond f$. $\Box f$ should be read as "it is necessary that *f*," and $\Diamond f$ should be read as "it is possible that *f*." Semantically, modal formulas are interpreted according to Saul Kripke's semantics, best known as Kripke frames. Basically, an interpretation in modal logic consists of a collection of many possible worlds or states. Pratt's discovery, further developed by other authors, led to the association of programs with modal operators. As a result, program logic can now make use of the well-developed corpus of modal logic.

Briefly, the dynamic logic approach to program logic is as follows: The association of a modal operator, [] and <>, with a program *P*, gives birth to the operators [*P*] and <*P*>. The exogenous characteristic of dynamic logic is now clear. If *f* is a formula (propositional or first order), then [*P*]*f* should be read as "necessarily, halting executions of *P* result in a state satisfying *f*." However, <*P*>*f* should be read as "possibly, halting executions of *P* result in a state satisfying *f*." Therefore, Hoare's logic statements such as {*a*}*P*{*b*}, in dynamic logic are expressed as $a \rightarrow [P]b$. Actually, dynamic logic subsumes Hoare logic and temporal logic as well. The semantics of dynamic logic are based on Kripke frames, demonstrat-

Copyright © 2002, American Association for Artificial Intelligence. All rights reserved. 0738-4602-2002 / $2.00



Network and Netplay: Virtual Groups on the Internet

Edited by Fay Sudweeks, Margaret McLaughlin and Sheizaf Rafaeli
Foreword by Ronald Rice

The vast, international web of computer networks that is the internet offers millions of users the opportunity to exchange electronic mail, photographs, and sound clips; to search databases for books, CDs, cars, and term papers; to participate in real-time audio-and video-conferencing; and to shop for products both virtual and physical. This huge conglomerate of links, hyperlinks, and virtual links is not just a technology for linking computers—it is a medium for communication. The convergence of computer and communication technologies creates a social convergence as well. People meet in chat rooms and discussion groups to converse on everything from automechanics to post-modern art; networked groups form virtually and on-the-fly, as common interests dictate. Like interpersonal communication, the networks are participatory, their content made up by their audience. Like mass-mediated communication, they involve large audiences. But the networks are neither purely interpersonal nor purely mass—they are a new phenomenon.

Network and Netplay addresses the mutual influences between information technology and group information and development, to assess the impact of computer-mediated communications on both work and play. Areas discussed include the growth and features of the internet, network norms and experiences, and the essential nature of network communications.

6 x 9, 320 pp. ISBN 0-262-69206-5

To order, call 800-356-0343 (US and Canada) or (617) 625-8569.
Distributed by The MIT Press, Cambridge, MA 02142

ing that dynamic logic is built on solid modal logic foundations.

Here we have a book that provides a deep insight into the topic of dynamic logic. However, readers of this magazine should be warned: This book does not provide tips on how to apply the concepts of dynamic logic to AI because the main focus of the authors is the use of dynamic logic as a formal system for reasoning about programs.

This 460-page book is divided into three parts: (1) fundamental concepts, (2) propositional dynamic logic, and (3) first-order dynamic logic. The first part provides readers with the necessary background to understand dynamic logic and makes the book self-contained. Despite the introductory aim of this part, its contents are rather deep, amounting to one-third of the book. This first part covers mathematical preliminaries, computability, complexity, logic, and reasoning about programs. Also, the authors provide an introduction to other topics related to dynamic logic, such as temporal logic, process logic, and Kleene algebra (but, strangely enough, these topics are covered in the last chapter of the book). The second part introduces propositional dynamic logic, covering syntax, semantics, properties, completeness, complexity, and so on. The third part, on first-order dynamic logic, is the most involved part of the book and introduces syntax and semantics, uninterpreted and interpreted levels, complexity, axiomatization, expressive power of languages, and so on.

This book is a comprehensive source of information on dynamic logic. It is aimed at researchers, teachers, and students of the subject. The book can be used in a dynamic logic course because all chapters come with exercises that teachers will find useful. If you are interested in program logics and program verification using dynamic logic, this is your book. Do not expect to find any information on the application of dynamic logic to AI or knowledge representation. However, AI researchers who want to deepen their understanding of the capabilities and limits of dynamic logic will find useful information in the book.

References

Allen, J. 1984. Toward a General Theory of Action and Time. *Artificial Intelligence* 23:123–154.

Dijkstra, E. W. 1968. Go To Statement Considered Harmful. *Communications of the ACM* 11(3): 147–148.

Fensel, D. 1995. *The Knowledge-Based Acquisition and Representation Language* KARL. New York: Kluwer Academic.

Hoare, C. A. R. 1969. An Axiomatic Basis for Computer Programming. *Communications of the ACM* 12(10): 576–580, 583.

McDermott, D. 1982. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science* 6(2): 101–155.

Meyer, J.-J.Ch. 1999. Dynamic Logic Reasoning about Actions and Agents. Paper presented at the Workshop on Logic-Based Artificial Intelligence, 14–16 June, Washington, D.C.

Moore, R. C. 1990. A Formal Theory of Knowledge and Action. In *Readings in Planning*, eds. J. F. Allen, J. Hendler, and A. Tate, 480–519. San Francisco, Calif.: Morgan Kaufmann.

Pratt, V. R. 1976. Semantical Considerations on Floyd-HCare Logic. In Proceedings of the Seventeenth Symposium on the Foundations of Computer Science, 109–121. Washington, D.C.: IEEE Computer Society.

van Harmelen, F., and Balder, J. R. ML^2: A Formal Language for KADS Models of Expertise. *Knowledge Acquisition* 4(1): 127–167.

Vilain, M. 1982. A System for Reasoning about Time. In Proceedings of the Second National Conference on Artificial Intelligence, 197–201. Menlo Park, Calif.: American Association for Artificial Intelligence.

Andrés Silva is assistant professor at Universidad Politécnica de Madrid. He started his research career at the Laboratory of Artifical Intelligence at Universidad Politécnica de Madrid and with the Group of Artificial Intelligence and Learning (GRAIL) at Universidad Carlos III. In 1998 to 1999, he worked at the Joint Research Centre of the European Commission in Ispra (Italy). He holds a Ph.D. in computer science from Universidad Politécnica de Madrid. His e-mail address is asilva@fi.upm.es.