The Knowledge Engineering Review, No. 7:1, 1992, 79–83

Book reviews

A new guide to artificial intelligence reviewed by Steve Easterbrook, School of Cognitive and Computing Sciences, University of Sussex, UK.
Formal methods in artificial intelligence reviewed by Paul J Krause, Imperial Cancer Research Fund, UK.
Formalism in AI and computer science reviewed by Paul J Krause, Imperial Cancer Research Fund, UK.
Formal techniques in artificial intelligence: a sourcebook reviewed by Paul J Krause, Imperial Cancer Research Fund, UK.

A new guide to artificial intelligence by Derek Partridge, Ablex, NJ, 1991, pp 546, £19.95. ISBN 0-89391-607-2

Many AI texts already exist, some good, some not so good, so why read this one? Partridge anticipates this question in his preface, and heads it off by carving out a new niche for himself. Too many introductory textbooks "skip across a few possibilities and leave it at that" (p. xvii). On the other hand the "genuine" texts, from established books such as Winston and Rich, to newcomers such as Tanimoto, simply provide a mass of detail about successful programs and popular mechanisms, while ignoring inadequacies and known problems, and failing to give a broader perspective. The difference about this book, so the preface assures us, is that it provides a survey detailed enough to introduce a broad and critical view of AI. It is presented as a companion to the current AI literature.

Attempting to assess how well the book lives up to this promise is difficult, because it is not clear who exactly the book is aimed at. Seen as purely a companion to the literature, the book is impressive: the bibliography is comprehensive, and there is a wealth of detail about the literature on a broad spectrum of topics from heuristic search to neural nets, from the details of Lisp and Prolog to the issue of AI methodology, from the nature of human intelligence to the limitations of studying toy domains. And herein lies the first problem: the book is difficult to read without access to this literature, whether by prior knowledge of it, or by diligent work in the library. Frequently, the book provides too little background to appreciate a point, referring instead to the appropriate source.

Judging by the tone of the book, the author is addressing a newcomer to the area. There are frequent attempts to enlist the reader as a sceptical outsider who can join the author in observing and mocking the endeavours of AI. While a healthy scepticism is no bad thing, the style in which it is done soon becomes irritating. For example, in the chapter on AI programming languages, recursion is presented as something the reader couldn't possibly have come across before, let alone understood: "You or I would program the function to compute the length of a list as a loop . . ." (p. 101). This chapter manages to both insult the intelligence of the reader, while completely failing to explain the terms used sufficiently for a novice to follow the cynical critique. In fact, the author is not above directly insulting his audience: ". . . if this sentence is more mysterious than usual, then you've skipped Chapter 1 or just failed to absorb at least one of the crucial bits" (p. 23).

I doubt the book would be much use to someone new to the field. The text is turgid and lacking in structure, overloaded with jocular asides and in-jokes, and littered with unhelpful forward references. Indeed, the author apologises for the forward references at more than one point. The early chapters of the book contain some detailed criticisms of the methodology and paradigms used

in AI, and of the Symbol System Hypothesis, in which explanatory material needed to follow the arguments is left until later sections of the book. But trying to locate such explanatory material is not easy, as many of the forward references add nothing to the argument. It is as if the author is so indecisive as to be unable to expunge any material in the interest of improving the structure. The result is that unexplained terms are introduced at inappropriate moments. For example, when discussing planning systems, this reference is irrelevant to the point being made: ". . . They propose to use a 'blackboard' architecture, which is not an edifice composed of old chalkboards, as you will find out if you read through chapter 6" (p. 50).

If the book is of no use as an introductory textbook, does it provide a useful critique for the seasoned hacker? The book is certainly full of critical discussion. But this discussion is clouded by a cynicism which interferes with clear exposition. Work in AI is demolished and ridiculed as it is described, and even, in places, before it is described. This is done in such a way as to obscure the nature of the subject. Rather than a clear presentation and critique of each area, the reader is presented with a flow of names, dates, histories and techniques, mingled with failings, put-downs and prejudices. Furthermore, some of the arguments are just plain wrong: chapter one ends with the point that because Samuel's checker playing program could consistently beat its designer, this must mean that the old adage that computers only do what they are literally told to do no longer holds. But this is no more a refutation than pointing out that a database can retrieve facts that the programmer had forgotten!

Finally, I must address the area that interested me in this book in the first place. I have been following Derek Partridge's ideas on software engineering and AI for some years, and was interested in what this book would have to say on the matter. It is encouraging to see an early chapter devoted to this area, and in particular the lack of attention to methodology in construction of AI programs. The argument presented runs like this: AI people do not exactly see eye-to-eye with computer scientists, and AI programs are hacked together without any sound methodology principles. Good practice in computer science demands a specify-and-verify approach, in which a formal specification is essential; indeed, software engineering can be equated with formal methods. AI problems are different because they are ill-structured and open-ended, and require an incremental, exploratory approach. Hence software engineering methods do not apply, and AI and software engineering have nothing to offer one another: AI must forge its own methodology.

This argument, whilst making some important points, is based on some fundamental misconceptions. Firstly, Partridge confuses science with engineering. To demand that programs built to demonstrate particular ideas, for example as PhD projects, should be properly engineered is to miss the point of the exercise. Such programs are not intended to be *used*. They exist to test the feasibility of scientific theories. Or more likely, they are inventions or tools which we might examine to develop scientific theories: theory usually lags well behind invention. Engineering is concerned with the production of real artefacts for real use, and hence requires rigour to ensure that design is based on sound principles. The point is that real, commercial systems are not feasible until the principles are understood. As an exploratory science, AI is not concerned with the production of robust systems. It does, however, occasionally produce robust (but limited) principles and techniques which can be used in the engineering of robust systems.

Secondly, the argument rests on a naïve and simplistic view of the nature of software engineering. In fact, Partridge plays fast and loose with the terms "computer science" and "software engineering", using them interchangeably. Theoretical computer science concerns itself with (among other things) the nature of computation. This includes reasoning about programs. In contrast, software engineering is concerned with how to design and build real systems. While this may involve application of techniques from theoretical computer science where appropriate, it certainly does not claim that problems which cannot be formally specified are "not appropriate for computerization" (p. 27).

After all, AI does not have a monopoly on ill-structured problems. Lehman (1980) characterizes three types of program: S-type (specifiable) in which the specification is the sole, definitive determinant of correctness; P-type, which are created to solve some stated problem, and success is

judged according to the problem statement; and E-type (embedded) which solve a problem or implement an application in some real world domain, and are judged according to acceptability, value and level of satisfaction. Large scale systems are nearly always E-type, and therefore the notion of correctness does not apply. Interactive systems are another important group of E-type programs, and methodologies have been developed which require an initial theoretical model, and follow an iterative cycle of implementation and evaluation (e.g., Boehm, 1988; Carroll et al., 1991; Giddings, 1984; Sharples et al., 1989). Note that such methodologies differ from Partridge's Run-Understand-Debug-Edit (RUDE) approach in both the initial, empirically validated model, and the emphasis on a stage of empirical evaluation in the cycle. What is important in such methodologies is not that the problem be formally specifiable, but that the requirements be explicitly and precisely stated, and furthermore, that they be updated as understanding of the problem changes. Formal methods can be applied to assist in the rigourous analysis of these requirements, but are not the be-all and end-all of software engineering.

To summarize, on the plus side this book is a compendium of useful information about AI. It attempts to provide a balanced critique, in that for all the criticisms, AI is still presented as a worthwhile endeavour. However, the book is poorly structured, and poorly written, and so is irritating to read, and difficult to use as a reference book. Furthermore, some of the arguments are misleading, and some are misconceived. In particular, the chapter on methodology takes a particularly simplistic and caricatured view of computer science, as an excuse to present a dubious methodology which takes no account of recent work in software engineering.

References

Boehm, BW, 1988, "A spiral model of software development and enhancement" *IEEE Computer* **21** (5): 61–72.

Carroll, JM, Kellog, WA and Rosson, MB, 1991, "The task-artefact cycle" in: JM Carroll, ed., *Designing Interaction: Psychology at the Human-Computer Interface.* Cambridge University Press.

Giddings, RV, 1984, "Accommodating uncertainty in software design" *Communications of the ACM* **27** (5): 428–434.

Lehman, MM, 1980, "Programs, life cycles, and laws of software evolution" *Proceedings of the IEEE* **68** (9): 1060–1076.

Sharples, M, Goodlet, JS and Pemberton, L, 1989, "Developing a writer's assistant" in: N Williams and P Holt, eds., *Computers and Writing.* Intellect.

Formal methods in artificial intelligence by Allan Ramsay, Cambridge University Press, Cambridge, 1991, pp 289, £14.95 (paperback). ISBN 0 521 42421 6.

Formalism in AI and computer science by Philip Leith, Ellis Horwood, Chichester, 1990, pp 225, £29.95. ISBN 0 13 325549 2.

Formal techniques in artificial intelligence: a sourcebook by RB Banerji (Ed.), North Holland, Amsterdam, 1990, pp 437, Dfl 160.00. ISBN 0 444 88130 1.

A father and his son were driving to a football match. As they were crossing a level crossing the car stalled. The lights warned of an approaching train. The father frantically tried to restart the engine, but in his panic had flooded the carburettor and the car was hit by the train. An ambulance was called to the scene within minutes, but on the way to the hospital the father died. The son was just alive when the ambulance reached the hospital, but his condition was very serious and he needed immediate surgery. He was wheeled into the emergency operating room and the surgeon summoned. On seeing the boy, the surgeon blanched and stammered "I can't operate on this boy—he's my son".

I have paraphrased this puzzle from Douglas Hofstadter's book *Metamagical Themas.* He asks the reader to resolve the puzzle, if indeed there is a puzzle, bearing in mind that the surgeon is not lying, and there are no tricks of reincarnation or adoption or whatever. It is quite a well known