Does AI have a methodology which is different from software engineering?

D. Partridge & Y. Wilks
Computing Research Laboratory, New Mexico State University
Box 3CRL, Las Cruces, NM 88003, USA

Abstract This paper argues that the conventional methodology of software engineering is inappropriate to AI, but that the failure of many in AI to see this is producing a Kuhnian paradigm 'crisis'. The key point is that classic software engineering methodology (which we call SPIV: Specify-Prove-Implement-Verify) requires that the problem be capable of being circumscribed or surveyed in a way that it is not, for areas of AI, like natural language processing. In addition, it also requires that a program be open to formal proof of correctness. We contrast this methodology with a weaker form complete Specification And Testability (SAT — where the last term is used in a strong sense: every execution of the program gives decidably correct/incorrect results) which captures both the essence of SPIV and the key assumptions in practical software engineering. We argue that failure to recognize the inability to apply the SAT methodology to areas of AI has prevented development of a disciplined methodology (which is unique to AI and which we call RUDE: Run-Understand-Debug-Edit) that will accommodate the peculiarities of AI and also yield robust, reliable, comprehensible, and hence maintainable AI software.

Introduction: Kuhnian paradigms in AI

Is it helpful or revealing to see the state of AI in, perhaps over-fashionable, Kuhnian (Kuhn, 1962) terms? In the Kuhnian view of things, scientific progress comes from social crisis: there are pre-paradigm sciences which struggle to develop to the state of “normal science” in which routine experiments are performed within an overarching theory that satisfies its adherents, and without daily worry about the adequacy of the theory.

At the same time, there will be other scientific theories under threat, whose theory is under pressure from either disconfirming instances or fundamental doubts about its foundations. In these situations, normal science can continue if the minds of adherents to the theory are closed to possible falsification until some irresistible falsifying circumstances arise, by accretion or by the discovery of a phenomenon that can no longer be ignored.
There is much that is circular in this (the notion of “irresistible” for example) and there may be doubts as to whether AI is fundamentally science or engineering (we return to this below). But we may assume, for simplicity, that even if AI were engineering, similar social descriptions of its progress might apply (see Duffy, 1984).

Does AI show any of the signs of normality or crisis that would put it under one of these Kuhnian descriptions, and what would follow if that were so? It is easy to find normality: the production of certain kinds of elementary expert system (ES) within commercial software houses and other companies. These work well enough for straightforward applications, yet doubts about their extensibility are widespread.

Crisis and pathology are even easier to find, and our diagnosis, in brief, is this: normal AI is impeded by the fact that, whether they are aware of it or not, a wide range of AI’s academic practitioners are struggling to conform to another paradigm because they suspect their own is inadequate. In our view the natural, normal, paradigm of AI is RUDE (Run-Understand-Debug-Edit). But pressure and crisis come from the SPIV (Specify-Prove-Implement-Verify) methodology, and its weaker version SAT (complete Specification-And-Testability of program behaviour). The nature of this crisis is not one of disconfirming instances — for how could that be, a factor which adds to the strong evidence that we should be talking in terms of engineering not scientific practice — but from pressure concerning foundations.

The basic pressure is derived from the methodology of software engineering (SE) and its unlikely allies, and their belief that software development must proceed by a certain path: that of SPIV. Work in expert systems, at least of the more simple-minded variety, is a leading factor in this pressure on AI, because it shares the central SPIV assumption that applications are, or should be, in areas of phenomena that can be specified completely with respect to their behaviour, and in advance of, not during, the process of programming. We shall discuss this issue in detail below; here we just wish to note a key ally of SPIV, and one that might be thought historically unlikely: Chomskyan linguistics and its current phrase-structure grammar successors (e.g. Gazdar, 1985).

The natural language case is a central and relevant one, it is the area of human phenomena modelled by AI where the strongest case can be made for data which are not of a type that allows complete pre-specification, in the sense that that would be the case if the set of sentences of, for example English, were a decidable set. Yet, Chomsky’s intention was always to show that his grammars did cover such a set (Wilks, 1986), and even though that enterprise failed, his successors have made it a central feature of their claims about grammar that the set to be covered should be recursively decidable (Gazdar, 1985). In the sense under discussion, therefore, recent work in AI and natural language processing (NLP) has been an example of SPIV methodology and in an area where it is, to some at least, the most counter-intuitive. We shall expand on this point below.

Our claim, then, is that AI methodology is under threat from an opposing paradigm, one not appropriate to AI’s subject matter, and one that encompasses conventional SE plus much of current ES and areas of NLP. As we shall discuss