Journal of Consciousness Studies, **2**, No.1, Spring 1995, pp. 52–66

CONSCIOUSNESS AS AN ENGINEERING ISSUE, PART 2

Donald Michie,[1] *University of Edinburgh, UK*

Abstract: This paper's first part (*JCS*, **1**, pp. 182–195), reviewed attempts to model real-world problem solving as machine-executable logic. Part 2 considers an alternative model in which the solution of problems is primarily the work of visualization supported by automatized skills. Consciousness operates at the level of goal-setting and monitoring, and of the construction and communication of after-the-event commentaries, not as a problem solver.

Engineering designs based on this model have proved convenient and effective. 'Structured induction' is now routinely used to recover and articulate expertise that in the human solver remains tacit. A seminal case of computer-mediated superarticulacy is described in which a human problem solver was enabled to transform an elaborate, but largely blind and unconscious, mental skill into a fully aware, communicable and machine-executable theory.

I: Introduction

Part 2 of this paper considers those processes that human problem solvers are least able to articulate, being for the most part quite unaware of them. Symbolic machine learning can recover models of such subcognitive routines, yielding the phenomenon of superarticulacy (Michie, 1986). Expert problem solvers, unable to articulate the detail of their own solving mechanisms, can readily exemplify them as recorded demonstrations. By application of learning algorithms to these records, a practised knowledge engineer can effect terse and perspicuous reconstructions of the previously mysterious skill. Moreover he or she can reproduce it dynamically by running the reconstructed procedures on the machine. In the strictly operational sense of articulating and discussing their own reasoning, their processes are more 'conscious' than a human thinker's could possibly be. The paper closes with one academic and one applied illustration.

II: Intelligence, Games and Robots

When a car's driver follows a familiar route while holding a conversation with a passenger, deliberative and reactive modes run side by side. If the conversation is more than small talk, the driver will recall its content rather than the driving. Yet the driving may have left its own imprint, only to be discerned when the same route is traversed, not as consciously accessible memories but as modified behaviours — particular caution at a given intersection, sharpened perception of certain road-signs, improvement in the smoothness with which a tricky bend is rounded, and so forth. This type of imprint has almost nothing to do with conscious awareness. As we shall see, it has almost everything to do with practical problem solving.

Since the early days of AI (see for example McCarthy, 1959) two forms of knowledge have been distinguished, corresponding roughly to Ryle's (1949) 'knowing that' and 'knowing how'. A similar distinction emerges from clinical studies of cognition and the brain. Summarizing effects of brain lesions on memory, L.R. Squire (1987, Ch. 11) defines *declarative* memory as 'memory that is directly accessible to conscious

[1] Present address: 73860 Shadow Mountain Drive, No. 2, Palm Desert, CA 92260, USA. This work was completed while the author was a Visiting Fellow in the School of Computer Science and Engineering at the University of New South Wales. Thanks are due to Professor John Hiller, Head of School, and to Professors Paul Compton and Claude Sammut for the hospitality and facilities of the School's AI Laboratory. I also thank Dr Michael Bain and Mrs. J.E. Hayes Michie for helpful criticisms and suggestions.

recollection'. He relates learned skills such as driving to separate *procedural* memories. The former acts as a repository for facts and events, the latter for procedures which, once thoroughly instilled by practice, are tacit.

Clones and profiles

Faced with an imaginary contract to develop an *articulate* robot driver, should the engineer implement separately the stimulus-response component of driving, or try to adapt mechanisms primarily developed for deliberative communication? In the construction of skilled artifacts such as aircraft autolanders, separatism has prevailed. Efficient implementations have mostly by-passed AI techniques. The results are at once task-efficient and opaque. When users demand transparency, retro-fitted 'explain modules' can be considered. An alternative method, 'behavioural cloning', uses symbolic learning to construct transparent rule-based implementations from an expert performer's selected stimulus-response data. The rules then not only guide the skilled program in emulating the expert, thus in some sense constituting a 'clone'. They also specify the implicit logic of the behaviour's subcognitive procedures, thus serving as a self-articulate 'profile'. There is no aim here of implementing neural circuitry, but only of transparently modelling its logic in a form which also efficiently implements the behaviour on a digital computer, while conferring a bonus of 'superarticulacy' (Michie, 1986), lacking from the tacit skill of the human exemplar.

Game-playing

When opacity is no bar, AI can indeed be by-passed. Mastery of board games has yielded to performance-first implementations that combine processor-intensive with store-intensive 'brute force'. Virtually nothing of intelligence accompanies the modern chess machine's gigabyte look-up tables of openings and end-game positions, and million-move-per-second search algorithms. To say this, however, carries an obligation also to say what one means by intelligence in board games such as chess.

Before the turn of the century Alfred Binet (1893) had already established the dominance of visual thinking in chess mastery. De Groot (1965) and Chase and Simon (1973) further demonstrated that the master's seemingly prodigious powers of memory are based on the mental retrieval and combination of stored patterns. Measurements and calculations by Simon and Gilmartin (1973) and by Nievergeldt (1977, 1991) independently showed that the size of the mental store (in the range 50–100,000) far exceeds the size of the consciously accessible pattern vocabulary, thus locating these patterns in the realm of subcognition. Manifestations of report-time chess intelligence, through commentaries and discussions of past games, receive passing attention from de Groot, but have not been followed up in depth. This mode is conspicuous by its absence from today's chess programs. The world's strongest programs defeat master players, but cannot communicate with their opponents, teach them or learn from them.

There is a further dimension, concerned with multiplicity of representation. In Minsky's (1994) words:

> If you understand something in only one way, then you really do not understand it at all . . . The secret of what anything means to us depends on how we have connected it to all the other things we know. That is why, when someone learns 'by rote,' we say that they do not really understand.

These criteria — whether embedded in tests of the communicability or of the multiplicity of internal models — are sufficient to disqualify claims to intelligence on behalf of the world's strongest game-playing programs, along with most other problem-solving programs. That this in no way disables them from world-class performance has repeatedly been demonstrated. Thus the world chess champion, Garry Kasparov, last year lost to a computer program in the quarter finals of a strong grandmaster tournament. In checkers during the same year the title of world champion passed to a program. As elsewhere detailed (Michie, 1995a), these programs are incapable of understanding or explaining what they are doing. They owe little to their meagre stores of conceptualized knowledge. On human criteria of awareness and content, they can justly be characterized as high-speed morons.

Will further evolution of game-playing programs continue along this same line? In the unlikely event that sheer performance continues to drive game-machine design then the answer is 'yes'. Otherwise it is 'no'.

The birth of each new technology opens a performance-first epoch in which engineering values dominate design. Aviation in the 1920s was spurred by aerobatic displays and by competitive events such as the Schneider trophy. Track racing and the land speed record played a similar part in the story of the automobile. But as these technologies matured and acquired widening circles of non-expert users, the engineer's dominance yielded to the pull of the market. Likewise in the twenty years since the first World Computer Chess Championship (machine-against-machine), a fast-growing consumer industry has begun to supply chess machines and chess programs to the mass consumer. But the new market is losing its appetite for sheer performance, already more than adequate. So manufacturers are re-grouping around user demand for intermingled chess lore, informative commentary, illustrative principles and worked examples. The demand is for articulate coaches, credibly aware of both users and self. Similar shifts are already visible throughout the information industries as a whole, particularly where (as in game-playing) decision-taking is interactive. Articulate intelligence is now in demand, but as a pressing cultural afterthought rather than to help the system perform better. As background to what follows, let us entertain the idea that the human brain's evolution traced a similar course.

Reactive behaviours

Numerous skills, from tying one's shoelaces and touch-typing to the generation and recognition of linguistic and musical expression, are found to be largely procedural and tacit (*see* Posner 1973, Anderson 1990, for overviews). In other cases the criteria by which choices are made may be declarative in nature, yet are still represented in the subjects' verbal reports by confabulations (Nisbett and Wilson, 1977; Dennett, 1992). It would be a mistake to take 'confabulation' here as necessarily derogatory or dismissive. Nisbett and Wilson cite Henry James's testimony to the ability of subcognitive work to impart artistic substance to an initially sketchy form:

> I was charmed with my idea which would take however much working out, and because it had much to give, I think, must have dropped into the deep well of unconscious cerebration: not without the hope, doubtless, that it might eventually emerge from that reservoir . . . with a firm irridescent surface and a noticeable increase of weight.

A possible metaphor for articulate consciousness is not so much the home movie maker as the creative director, integrating as best he can the torrent of offerings from unseen armies — script-writers, cast, cameramen, rewrite men, sound-recorders, dubbing editors, filing clerks and the rest — to whom for some reason he enjoys no direct access.

Other forms of conscious thought, that we shall term ‘para-articulate’, proceed primarily through the associative manipulation of images in the ‘mind’s eye’ (Kosslyn, 1980, 1983). Experimental and theoretical wings of AI both acknowledge graphical forms of declarative representation, and are aware (sometimes uncomfortably) of the dominant role in mental expertise of para-articulate and subarticulate components. Some early AI studies were motivated by attempts to bring machine equivalents of the mind’s eye and of the mind’s logical calculator to converge on selected problems, in a style resembling what we as human solvers surely do ourselves. Outstanding among these was Gelernter’s (1959) approach to machine-generated proofs in elementary Euclid. His program used diagrams to supplement its symbolic reasoning process, chiefly by closing off directions of potentially time-consuming search that graphical considerations showed to be unpromising. Euclidean geometry holds a seminal place in European intellectual history, yet neither Gelernter’s demonstration, nor the principle illustrated, has been seriously followed up. Thirty years of diverse achievements leaves the AI field still unequipped even for a Turing Test restricted to trivial subsets of plane geometry!

For better or worse, then, mainstream AI has remained committed to what Nilsson (1994) describes as the standard approach ‘based on explicit declarative representations and reasoning processes’. Among alternative approaches he cites ‘the so-called *behaviour-based*, *situated*, and *animat* methods (Brooks, 1986; Maes, 1989; Kaebling and Rosenschein, 1990; Wilson, 1991), which convert sensory inputs into actions in a much more direct fashion’. These alternative architectures are essentially data-driven, or *reactive*, as opposed to model-driven. Omitting intelligence, they limit themselves to the automatic solution of problems by structured repertoires of reflex-like responses. One should pause here to note that Nilsson (1994; also Benson and Nilsson (1995)) has a new design that combines a top-level goal-driven layer with a purely reactive bottom level. T-R systems, as they are termed, have shown up well under initial tests.

For the moment, though, our concern is with what can be accomplished through the interaction of numbers of reactive systems in architectures entirely lacking the top level. Brooks’ intended application to space robotics deserves particular mention. The aim is to reduce complexity, cost and failure risks of planetary missions by employing many simple, small, insect-like robots as surface exploration vehicles. The key idea is to avoid requiring the robots to maintain complete three-dimensional models of their surroundings. ‘Instead, we layer many simple behaviours, each connecting sensors to actuators, to produce a robust ‘‘instinct’’ driven control system for the robots.’ According to Brooks (1994):

> there have been a number of Earth-based demonstrations of such robots carrying out simulated planetary integration with a flight vehicle, an actual soft landing, robot development, and returns of video data, while the six-legged robot explored the surrounds of its landing site.

Machine equivalents of conscious intelligence are hardly likely to emerge from straight-line evolution of such designs, although some have claimed that solving sufficiently hard computer-science problems is *ipso facto* ‘doing AI’. In line with the definitions of this review, Brooks distances himself from this claim. Indeed he explains that the letters ‘AI’ on his laboratory door stand for ‘Artificial Insects’. Brooks’ point is that an extremely hard computational problem is here solved *without* need of intelligence. Extending the point to modern chess machines, these masterpieces of hardware, software and database techniques are not to be seen as AI in action. Nor can there be any *necessary* requirement for embedded intelligence in mechanizations of chess mastery, unless the task is widened to include game-annotation or teaching, in addition to actual play. If chess machines, or for that matter Brooks' Mars robots, had to furnish a plausible rationale for decisions, then, and only then, would conceptual analysis become indispensable.

III: From Subcognition to Superarticulacy

In the last section Brooks was cited to the effect that performance-only systems have no compelling need for AI. So long as the argument remains purely academic, this position is hard to fault. But in the outside world the idea of 'performance' can include costs of field maintenance. A relevant tabulation is given in Michie (1991) of the trivial costs of maintaining knowledge-based software relative to those associated with software generally. Costs and difficulties of troubleshooting opaque systems may be likened to those that attend a physician's diagnostic examination of a deaf-mute patient unable to read or write. In software, AI introduces the option of adopting a self-articulate architecture in the first place. Further, clients may demand that a delivered system should perform not only efficiently but also transparently enough for the company's experts in the given domain to cross-question it.

Modern work along this line traces back to a discovery by Shapiro (Shapiro and Niblett, 1982; Shapiro and Michie, 1986; Shapiro, 1987), who combined classical principles of structured programming with the then relatively new art of mechanized inductive inference. The laboratory result was the machine-mediated articulation of an elaborate chess concept previously resident, subarticulately, in every master player. The technique escaped academic attention (Shapiro and his associates had meanwhile moved into industry) but found a world-wide niche in commercial software. Shapiro's 'structured induction' allows tacit forms to be recovered and given explicit expression. I have made available elsewhere relevant details of contemporary industrial use and of related aspects of tacit skills (Michie,1995b).

Structured induction: theory from intuition

The wave and corpuscular theories of light are differently constructed accounts of unchanging laws of optics which are not themselves constructed. Both theories are valid, but under some conditions the wave theory is of more help, while under other conditions the corpuscular theory offers more. But what of a case where two or more theories are demonstrably concise and true over the whole of a given defined domain, yet one is highly effective and the others unfit for human use? Until recently such a situation was difficult to envisage. Mechanizations of inductive inference, however, can now generate indefinitely many complete and correct theories to fit the same set of facts. Shapiro's computer-aided construction of alternative theories for the adjudication of a well-known but non-trivial chess end-game provided the first demonstration.

Shapiro considered the 209,718 legal positions in which White has the move with king and pawn versus king and rook, with White's pawn on a7 threatening to queen immediately. Figure1 illustrates such a position.

After varying amounts of deliberation of usually less than a minute's duration, chess masters can say with confidence, and with more than 99% accuracy, whether White can win or not. They cannot, however, say in detail how they do this. They can name only the top-level concepts employed, and even these they cannot define with precision. Shapiro's question was: given expert help in identifying the lowest-level patterns ('primitive attributes'), is it possible to hand-program these primitives and then to machine-construct from them a complete symbolic representation of the tacit theory?

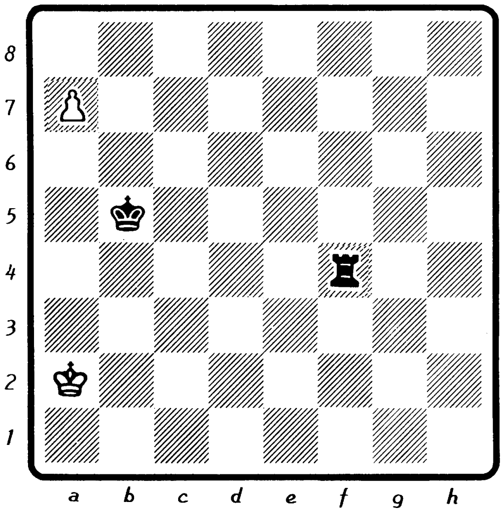


Figure 1: A chess endgame position with White to move. Is it won for White? To answer, the theory of Figure 3 considers whether Black can delay White's queening the pawn by means of a double-attack threat (DBLAT, level 3.4 in Figure 3). First it is established that the White king is on the edge and is not on a8, then that the Black king does not obstruct the rook's access to that edge, then that White's immediate promotion of the pawn would not attack the Black king and finally that the Black king controls the intersection point (a4) to which the rook threatens to move. From all of this it is concluded that the answer is 'no'. A chess expert familiar with this endgame leaps to such a conclusion at a glance, although usually capable of then reconstructing a justification by retrospective analysis.

He noted that:

1. no such representation had ever been constructed by human agency;
2. since the expertise is largely inaccessible to conscious review, no such representation, in the opinion of chess masters consulted, could ever be so constructed;
3. no complete representations (brain compatible or otherwise) could be constructed by known programming techniques other than by exhaustion of the domain;

- the BR can be captured safely (**rimmx**)
- one or more Black pieces control the queening square (**bxqsq**)
- there is a good delay because there is a hidden check (**hdchk**)
- there is a special opposition pattern present (**spcop**)
- the WK is one away from the relevant edge (**wtoeg**)
- the kings are in normal opposition (**dsopp**)
- the WK distance to intersect point is too great (**dwipd**)
- there is a potential skewer as opposed to fork (**skewr**)
- the BK is not attacked in some way by the promoted WP (**bkxbq**)
- the BR attacks a mating square safely (**rxmsq**)
- the BK can attack the WP (**bkxwp**)
- the mating square is attacked in some way by the promoted WP (**qxmsq**)
- the BR does not have safe access to file A or rank 8 (**r2ar8**)
- the WK is on square a8 (**wkna8**)
- B attacks the WP (BR in direction x = – 1 only) (**blxwp**)
- a very simple pattern applies (**simpl**)
- the WK is in stalemate (**stlmt**)
- the WK is in check (**wknck**)
- the BK can attack the critical square (b7) (**bkxcr**)
- the BR bears on the WP (direction x = – 1 only) (**rkxwp**)
- there is a skewer threat lurking (**thrsk**)
- B can renew the check to good advantage (**mulch**)
- the WK is on an edge and not on a8 (**cntxt**)
- the BK is not in the way (**bkblk**)
- the BK controls the intersect point (**katri**)
- the WK is in a potential skewer position (**wkpos**)
- the WK cannot control the intersect point (**wkcti**)
- the BK can support the BR (**bkspr**)
- the BR alone can renew the skewer threat (**reskr**)
- the WK can be skewered after one or more checks (**skach**)
- the WK can be reskewered via a delayed skewer (**reskd**)
- the BK is not in the BR's way (**bknwy**)
- the BR can achieve a skewer or the BK attacks the WP (**skrxp**)
- the BK is on file A in a position to aid the BR (**bkona**)
- the BK is on rank 8 in a position to aid the BR (**bkon8**)
- the WK is overloaded (**wkovl**)

Figure 2: List of primitive attributes invoked. The same set served for both the unstructured and the structured decision rules.

4. because of the inaccessibility of expertise to conscious review . . . the so-called ‘knowledge engineering’ techniques of contemporary expert systems work would not be applicable;
5. in the absence of information about the structural features that automatically generated rules should exhibit, conventional use of inductive learning techniques would be inadequate. Rules generated in this way might well be complete but not brain compatible.

In regard to Shapiro’s point 5, so it turned out. The human–computer co-operation to construct a ‘brain compatible’ theory eventually succeeded. A machine-generated database comprising a complete look-up table of the 209,718 positions was then constructed. 129,825 were found to be won for White, and 79,893 not won, corresponding in all but 15 cases with the theory’s adjudications. Using the resulting look-up table as a source of example decisions, conventional computer induction (Shapiro’s point 5 above) then condensed it into indefinitely many complete and logically equivalent ‘theories’ of solely machine authorship. These were then compared with the human–computer product.

The smallest of 50 complete and correct theories so generated comprised a decision rule involving 82 tests and outcomes. In the original publication it occupies over two pages with formless meandering. To interpret it requires the vocabulary of primitive attributes. These are tabulated in Figure 2, each definable as a simple geometrical pattern. The structure displayed in the Figure, with appropriate substitution of Figure 2’s vocabulary, now reads:

```
if the Black rook can be captured safely then won for White
  otherwise if one or more Black pieces control the queening square then not
                                                          won for White
    otherwise if the white king is in check then not won
      otherwise if the Black king is not attacked in some way by the promoted
                                               white pawn then not won for White
        otherwise if the White king is on square a8 then not won for White
          otherwise if the Black king is not in the way then not won for White
            otherwise if either king controls the intersect point then
                    if the Black king does then not won for White
                    if the White king does then won for White
              otherwise if the White king is in a potential skewer position then ...
```

and so on, through tests of a further 28 primitive properties, of which many have multiple invocations from different parts of the rule-structure.

What can one say about such a ‘theory’? First, whether run on the machine or humanly executed with paper and pencil, it gives the right answer to every question. Second, in contrast to its building materials (the primitive attributes), it is not humanly constructed. But for those who see theories as mental tools it is not a theory at all. As an explanatory instrument it is misshapen, repellent and unusable. No-one could rationally hope to

Figure 3 (opposite): The nine concepts that were machine-defined to constitute the above rule-set correspond to the nine white boxes of Figure 4, and the numbered levels to those of the diagram. The lower-case names in parentheses correspond to occurrences of the 35 primitive concepts (grey boxes in Figure 4) from which the rule-set derives its adjudication of each case according to the if–then–else logic expressed here as IFF’s, AND’s, OR’s and NOT’s. These logical expressions correspond to the elongated black boxes of Figure 4. Introductory sentences beginning ‘This rule is used . . .’ are explanatory interpolations by Shapiro (see text for reference and context).

PA7, top-level rule. This rule is used to decide if a KPa7KR position with White-to-move is won-for-White or not.

```
KPa7KR is won for White (PA7, 1) IFF
    the BR can be captured safely (rimmx)
    OR none of the following is true:
    there is a simple delay to White's queening the pawn (DQ, 2.1)
    OR one or more Black pieces control the queening square (bxqsq)
    OR the WK is in stalemate (stlmt)
    OR there is a good delayed skewer threat (DS, 2.2)
```

DQ, level 2.1 This rule is used in the PA7 (top-level) rule to decide if Black can successfully delay White from queening its pawn.

```
There is a simple delay to White's queening the pawn (DQ, 2.1) 1FF
    there is a good delay because there is a mate threat (THRMT, 3.1)
    OR there is a good delay because the WK is on square a8 (WKA8D, 3.2)
    OR there is a good delay because the WK is in check (WKCHK, 3.3)
    OR there is a good delay because of a double attack threat (DBLAT. 3.4)
    OR there is a good delay because there is a hidden check (hdchk)
```

DS, level 2.2 This rule is used in the PA7 (top-level) rule to decide if Black can force a double attack of type 'skewer'.

```
There is a good delayed skewer threat (DS, 2.2) 1FF
    there is a special opposition pattern present (spcop)
    OR all of the following are true:
    the WK is one away from the relevant edge (wtoeg)
    AND the kings are in normal opposition (dsopp)
    AND the WK distance to intersect point is too great (dwipd)
    AND there is a potential skewer as opposed to fork (skewr)
    AND the BK is not attacked in some way by the promoted WP (bkxbq)
```

THRMT, level 3.1 This rule is used in the DQ (level 2.1) rule to decide if Black can successfully delay White from queening its pawn because of an initial checkmate threat.

```
There is a good delay because there is a mate threat (THRMT, 3.1) IFF
    the BR attacks a mating square safely (rxmsq)
    AND
    EITHER the BK can attack the WP (bkxwp)
    OR none of the following is true:
    the BK is attacked in some way by the promoted WP (bkxbq)
    OR the mating square is attacked in some way by the promoted WP (qxmsq)
    OR the BR does not have safe access to file A or rank 8 (r2ar8)
```

WKA8D, level 3.2 This rule is used in the DQ (level 2.1) rule to decide if Black can successfully delay White from queening its pawn because the white king is initially on the pawn promotion square (a8).

```
There is a good delay because the WK is on square a8 (WKA8D, 3.2) IFF
    the WK it on square 8 (wkna8)
    AND any of the following is true:
    the BR has safe access to file A or rank 8 (r2ar8)
    OR B attacks the WP (BR in direction x = –1 only) (blxwp)
    OR a very simple pattern applies (simpl)
```

WKCHK, level 3.3 This rule is tested in the DQ (level 2.1) rule to decide if Black can successfully delay White from queening its pawn because the white king is initially in check.

```
There is a good delay because the WK is in check (WKCHK. 3.3) IFF
    the WK is in check (wknck)
    AND the BR cannot be captured safely (rimmx)
    AND any of the following is true:
    B can attack the queening square soon (BTOQS, 4.2)
    OR the BK can attack the critical Square (b7) (bkxcr)
    OR the BR bears on the WP (direction x = –l only) (rkxwp)
    OR there is a skewer threat lurking (thrsk)
    OR B can renew the check to good advantage (mulch)
```

DBLAT, level 3.4 This rule is used in the DQ (level 2.1) rule to decide if Black can successfully delay White from queening its pawn because of an initial double-attack threat.

```
There is a good delay because of a double attack threat (DBLAT. 3.4) IFF
    the WK is on an edge and not on a8 (cntxt)
    AND the BK is not in the way (bkblk)
    AND the BK is not attacked in some way by the promoted WP (bkxbq)
    AND
    EITHER the BK controls the intersect point (katri)
    OR
    the WK is in a potential skewer position (wkpos)
    AND the potential double attack is good (OKSKR, 4.1)
```

OKSKR, level 4.1 This rule is used in the DBLAT (level 3.4) rule to decide if Black can successfully capitalize on a potential double-attack threat.

```
The potential double attack is good (OKSKR, 4.1) IFF
    the BR has safe access to file A or rank 8 (r2ar8)
    OR the WK cannot control the intersect point (wkcti)
    OR the BK can support the BR (bkspr)
    OR the BR alone can renew the skewer threat (reskr)
    OR the WK can be skewered after one or more checks (skach)
    OR the WK can be reskewered via a delayed skewer (reskd)
```

BTOQS, level 4.2 This rule it used in the WKCHK (level 3.3) rule to decide if any Black piece will soon be able to control the queening square as a result of the white king having initially been in check.

```
B can attack the queening square soon (BTOQS level 4.2) IFF
    the BK is not in the BR's way (bknwy)
    AND any of the following is true
    the BR can achieve a skewer or the BK attacks the WP (skrxp)
    OR the BK is on file A in a position to aid the BR (bkona)
    OR the BK is on rank 8 in a position to aid the BR (bkon8)
    OR the WK is overloaded (wkovl)
```

internalize it for mental adjudication of positions. Finally, it has no discernible relation to the way in which a chess master thinks.

By way of contrast, Figure 3 contains the structured theory built co-operatively from the same vocabulary of 36 primitive concepts, supplemented with eight intermediate concepts identified and named by the chess master and implemented by machine. In Shapiro's final version the machine also rendered the whole into English.

The theory's 15 errors out of 209,718 cases was two orders of magnitude less than the chess master's error rate. Using a built-in self-commenting feature to be described, it was amended to eliminate all 15 errors in under two hours. The mode in which structured induction is performed is as follows.

The expert is shown a position and asked: 'What properties of this case do you need to know in order to adjudicate it?' A list is made from his answers, and incremented by posing the same question for further selected cases. When the list of position-attributes has settled down, no further progress can be made without machine aid. This is because once the half-dozen or so tests needed to classify a given case have been identified, the expert can give no further account that could be programmed either of how he applies each test or of how he combines the results into a decision procedure. At this point computer induction takes over the task of articulation.

Returning to Shapiro's historical case, the requirements list for the top-level concept (is the position won for White?) consists of (in alphabetical order of attribute names):

Requirements list for 'is the position won for White?'

- does one or more Black piece(s) control the queening square (bxqsq)?
- is there a simple delay to White's queening the pawn (DQ)?
- is there a good delayed skewer threat (DS)?
- can the Black rook be captured safely (rimmx)?
- is the White king in stalemate (stlmt)?

Of the above 5, three are simple enough to be implemented as hand-coded routines, typically as procedures of from 10 to 20 lines of C code. The other two, allotted upper-case names above, present difficult evaluation problems in their own right. They are declared to be subproblems and the process is iterated. Taking the more complex of the two, namely DS, the master is asked: 'What properties of a position do you need to know in order to decide whether there is a simple delay to White's queening the pawn?' Once more, successive confrontation with sample positions elicits a chain of responses that enables the knowledge engineer to converge on a stable list, as follows:

Requirements list for 'is there a simple delay to White's queening the pawn?'

- is there a good delay to queening because of a double attack threat (DBLAT)?
- is there a good delay because of a hidden check (hdchk)?
- is there a good delay to queening because of a mate threat (THRMT)?
- is there a good delay because the White king is on a8 (WKA8D)?
- is there a good delay because the White king is in check (WKCHK)?

A glance at the hierarchical diagram (Figure 4) corresponding to the theory of Figure 3 shows that, to cover some positions, iterative decomposition does not terminate until the level-4 subproblems, 'can Black attack the queening square soon (BTOQS)?' and 'is the potential double attack good (OKSKR)?', needing only primitives for their definitions.

When all terminal levels of the hierarchy have been reached the final phase proceeds bottom-up. Thus a machine-induced machine-executable recognizer for BTOQS above

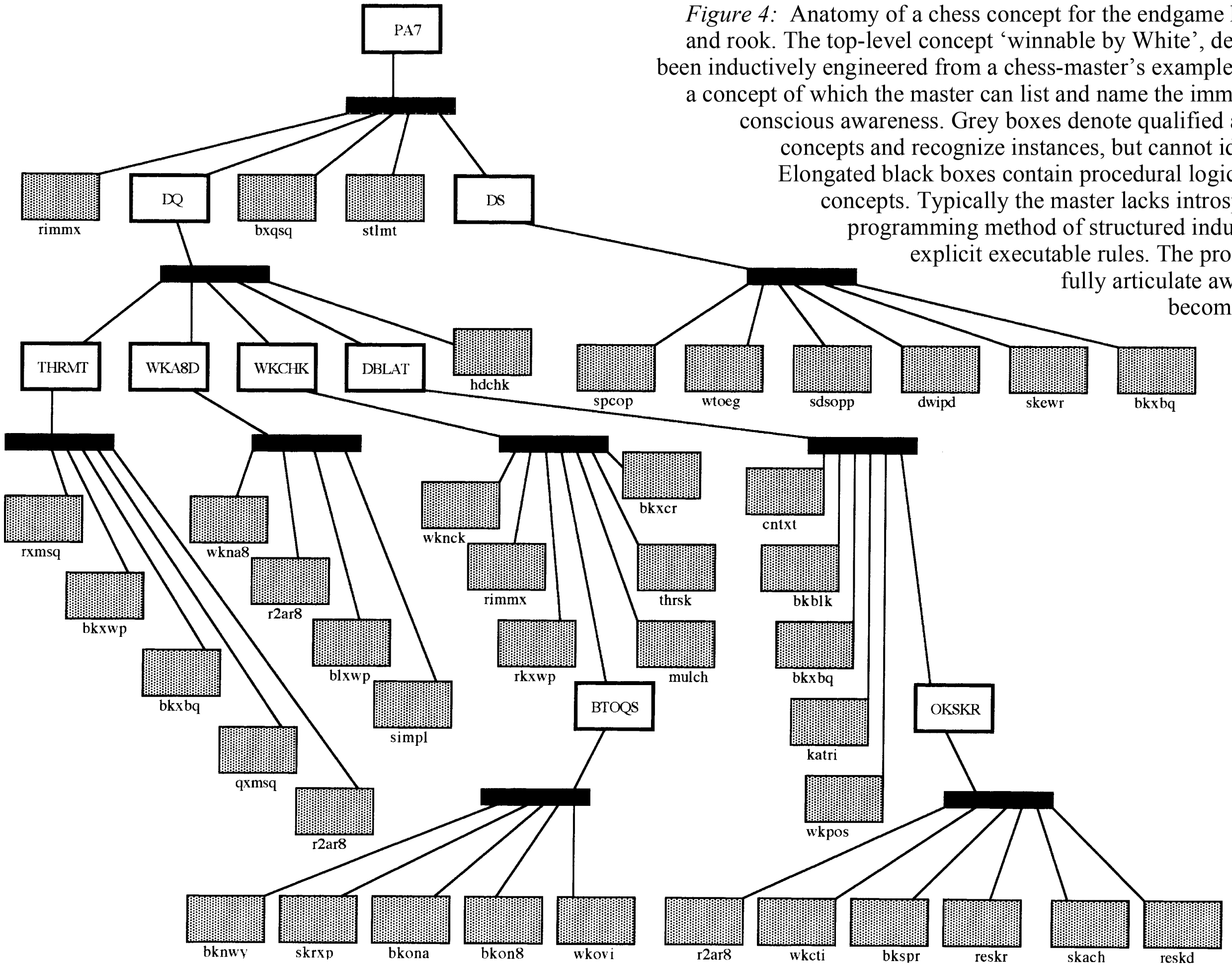


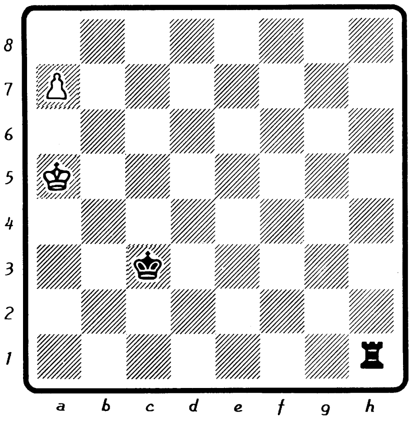
Figure 4: Anatomy of a chess concept for the endgame king and pawn (on a7) against king and rook. The top-level concept 'winnable by White', denoted by the box labelled PA7, has been inductively engineered from a chess-master's example decisions. Each white box denotes a concept of which the master can list and name the immediate subconcepts, thus exhibiting conscious awareness. Grey boxes denote qualified awareness: the player can name the concepts and recognize instances, but cannot identify more primitive constituents. Elongated black boxes contain procedural logic for recognizing the corresponding concepts. Typically the master lacks introspective access to this logic. But the programming method of structured induction reconstructs it in the form of explicit executable rules. The program thus acquires unqualified and fully articulate awareness. The master's black boxes become white in the reconstructed model. But the human's non-black boxes darken in use, becoming 'intuitive'.

is all that is needed to machine-induce a runnable recogniser for WKCHK (see Figures 3 and 4). In practice ways may suggest themselves of shortcuttting the construction of the needed recognizer. But under full mechanization a complete 'truth table' of specimen positions can be generated, each line exhibiting a different feasible combination of properties for the expert to 'fill in the blanks'.

Only one procedural (as opposed to primitive) property remains at level 4 to be similarly machine-expressed as a rule by the same method, namely OKSKR (see Figure 4). All is ready for explication of the four procedural properties at level 3, the two at level 2 and finally the top level. Everything is now available either for automatic generation of the English version of Figure 3, or for interactive use of the theory by submitting disputed positions for adjudication. The latter could equally be done by look-up from the database. The difference is that a theory-driven system can accompany its verdict on each case with a reasoned reply. Figure 5 shows the system's responses to the first and last of six questions of which the last was submitted from the floor at the fourth international Advances in Computer Chess meeting (Shapiro and Michie, 1986).

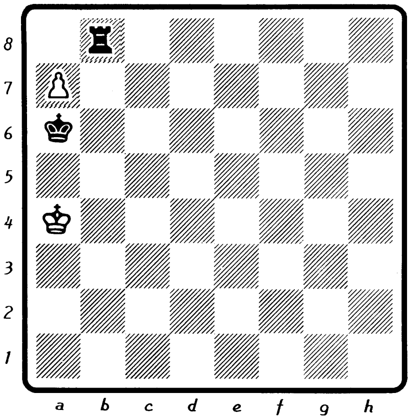
Partitioning expertise: levels of introspectability

When end-game specialists were allowed to converse with this system in a 'Turing Test' relaxation of the style of Figure 5, their impression was of contact with a sophisticated and resourceful mentality that exceeded their own grasp of the given micro-world at every level. How to model the difference between this well engineered king-pawn-king-



Position 1, COMMENT BY PROGRAM:

Since the Black rook has safe access to file *a* or rank 8 (because the White king can't cover both intersect points) it follows that the potential skewer is good; from this it follows there is a good delay due to a double attack threat; from this it follows there is a simple delay to White's queening the pawn; from this it follows that this position is not won for White.



Position 6, COMMENT BY PROGRAM:

Since the Black rook can be captured safely it follows that this position is won for White.

Figure 5: To give the flavour of the way in which the structured theory shown in Figure 4 can be made to yield reasoned justifications, the figure above reproduces the system's responses to the first and last of six representative cases for adjudication. The last case was a trick question thrown up at an international meeting. The idea was to trap it into a justification based on the pawn capturing the Black rook with promotion to queen — with the counterproductive effect of statlemate. Through the qualification 'safely', the system's answer covers the possibility of promotion to rook, which both avoids stalemate and wins. The response was accordingly judged adequate, although terse.

rook mentality and the less developed intellectual consciousness of human end-game specialists?

The sub-goal hierarchy of Figure 4 offers a model. Goals, represented in the diagram as white boxes, are declarative concepts that in the early stages of acquisition of a complex skill remain open to introspection. The grey boxes of the Figure represent primitive attributes at the hierarchy's bottom level. At intermediate levels elongated black boxes denote the if-then-else procedures that use lower goals to define higher ones. In human solvers these procedures are subcognitive, 'intuitive', non-introspectable. But in machine intelligences all the boxes can be introspected and their contents expressed symbolically. This superarticulacy enables the expert program to simulate a more 'conscious' grasp than any human exponent of the given domain. Moreover, as the latter perfects his or her expertise, so automatization spreads from lower to higher levels: white boxes successively become grey, and grey boxes blacken. Overall such progressive loss of conscious awareness represents net gain. In his 1911 book *Introduction to Mathematics* A.N. Whitehead pointed out that 'it is a profoundly erroneous truism . . . that we should cultivate the habit of thinking what we are doing. The precise opposite is the case. Civilisation advances by extending the number of important operations which we can perform without thinking about them.' Loss of communicability may be a small price to pay when we are seldom called on to explicate for others our accumulated intuitions. Yet we might like, if only we could, to eat our cake *and* keep it for social sharing. Structured induction opens a way to do just this. Figure 3 represents an illustrative crumb that can be digested by a human conscious intelligence — but only by virtue of its first having been articulated by an artificial one.

In inductively engineered representations of expertise, all boxes can fairly be represented as white. For expert humans most are black or eventually become so. The difference manifests itself in the potential for explicit social sharing of acquired expertise limited in humans, unlimited in machines. This is why when approaching consciousness as an engineering issue, specialists in structured induction can go beyond imitation of nature and seek to improve upon what they find. It should be re-emphasized that the term 'consciousness' here and throughout is interpreted in an operational sense mandatory for engineers — that is, in terms of observable manifestations rather than of sensations.

Mechanizing the discovery of global structure

The foregoing case concerns the computer-based discovery of classificatory theories. Techniques and software have migrated into the practice of half a dozen or so commercial suppliers of the 'expert system' niche market. But the human partner still carries more than a fair share of the burden of conceptualization. This impression is supported by calculation, which shows that eighty-five percent of the decision information was contained in the hierarchical structure, and only fifteen percent in the computer induced rules.

Can the discovery of hierarchical structure, together with associated intermediate concepts, be itself mechanized? Two current approaches have made different inroads into this problem. Gaines (1995) uses what he terms 'exception-directed acyclic graphs'. Starting with Shapiro's original 3196 attribute-vectors his program automatically constructed a theory which the expert eye has no difficulty in mapping to an economical and fully expressive subset of the original structured theory. The mapping appears straightforward, but until it has in fact been automated the potential importance of this advance remains *sub judice*.

A second prospect of escaping structured induction's confining limitations arises from a recent marriage of machine learning with logic programming known as Inductive Logic

Programming (ILP). The full expressive power of first order logic here combines with a natural approach to the automatic structuring problem via ‘predicate invention’.

These advances will, however, remain of limited interest until such time that conditions favour entry of computer theory-generation into a significant stream of human intellectual life. Such conditions are now arising in scientific specialisms where aids to informed conjecture are at a premium. In biomolecular modelling the conjecture of structure-activity relations has a critical bearing on the economics of drug design. Human-computer mental rapport at the level of shared scientific concepts is thus becoming both an urgent desideratum and a practical possibility.

Communicable laws

Substantial gains accompany extraction from biomolecular data of theories able to suggest new compounds for trial syntheses. Rules must not only predict with good accuracy the specified biological activity but should also make sense to the scientists themselves. The following example is taken from a summary of new applications of symbolic machine learning to structural molecular biology (Sternberg *et al*., 1994) and relates to the inhibition of *E. coli* dihydrofolate reductase by 2,4,-diamino-5-(substituted-benzyl) pyrimidines, to which a great deal of previous study had been devoted by others using numerical computations:

```
drug A is better than drug B if:
  drug A has a substitutent at position 3 with
    hydrogen-bond donor = 0 and
    π-acceptor = 0
    and polarity >0 and
    size <3 and
  drug A has a substituent at position 4 and
drug B has no substituent at position 5.
```

Sternberg and his co-workers summarize the approach used for the application of machine learning to scientific problems.

> There is an interactive cycle between human analysis and machine learning. Initially traditional methods process the data and develop representations that characterize the system and rules describing the relationship between the components of the system. Next machine learning uses these representations to identify new, and hopefully more powerful and incisive, rules. Then human intervention is required for interpretation of the rules and the cycle can be repeated.

For the method fully to come into its own, automated hierarchical problem decomposition needs to be introduced into the working context. In the Shapiro system this was performed by the expert with support from the system. As mentioned earlier the intrinsically more powerful algorithms of Inductive Logic Programming can in principle (but not yet in routine practice) incorporate ‘predicate invention’, i.e. machine discovery of intermediate concepts. Problem-decomposition is then shared more equally between human and machine partners. Efficient integration of this process into ILP’s theory-building software will extend its role from that of scientist’s assistant to that of junior colleague.

IV: Conclusion

No further departure of technique is required for the above development. We can foresee its being consolidated, and at the same time blended with the newest offerings of commercial ‘agent’ technology. Attribution of conscious thought to machine partners will become harder to resist. Turing spoke of a ‘polite convention’ whereby human thinkers ordinarily concede that other people too can think. Workers in busy laboratories and offices, perhaps initially only in absent-minded moments, will inevitably slip into the same convention towards their increasingly intelligent machine helpers.

The process has in fact already started — but with little indication yet that users take their idiom seriously. When we bring the operating system’s latest odd behaviour to the local system guru, we get something like:

> Well, it went to look for your application, but obviously had the wrong idea about which subdirectory it should look in. So it tried opening your file using the standard editor. When that didn’t work it jumped to the crazy conclusion that it was dealing with a compressed file. Now it wants you to decompress it. Try humouring it and see what happens.

The guru’s colleagues know that, put to the question, he or she will not maintain that the system *really* gets ideas, tries things, jumps to conclusions, has lunatic spells, etc.

The above point centres on inter-user discourse *about* the system. But discourse *between human and machine* is coming to the fore, revising the notion of ‘user’ in the direction of ‘partner’ or ‘colleague’. It is precisely on this point that commercially sponsored research in human–computer interaction is experimenting with nonverbal supports for human–computer ‘rapport’ — signalling through facial expression, tones of voice etc. To relate all this to consciousness we have to replace the unitary notion, with which we started, by its modern subdivision into ‘intra-subjective’ and ‘inter-subjective’ (see, for instance, Trevarthen, 1979). The former relates to self-awareness. The latter denotes the collective awareness of each other’s moods and mental experiences as these arise in a closely knit group. Such rapport is the basis of conveying and sharing adjustments of attitude, implicit negotiations, appeals to group standards and ideology, status considerations, and many other non-logical processes whereby intellectual consensus is sought. In mixed workgroups will these social arts of intimacy be confined to human–human exchanges during brainstorming sessions? Or will one day the brainstorm logs of mixed workgroups begin to include human–computer passages in which can be clearly discerned similar footprints of rapport?

When in this or some future century that day arrives, a more important test by far will have been passed than the one proposed by Alan Turing.

References

Anderson, J.R. (1990), *Cognitive Psychology and its Implications* (3rd edn., New York: W.H. Freeman).

Benson, S. and Nilsson, N. (1995), ‘Reacting, planning, and learning in an autonomous agent’, in *Machine Intelligence 14*, ed. K. Furukawa, D. Michie and S. Muggleton (Oxford: Clarendon Press), in press.

Binet, A. (1893), in *Revue des Deux Mondes*, **117**, pp. 826–59. English translation published as ‘Mnemonic virtuosity: a study of chess players’, *Jour. Genet. Psychol.*, **74**, pp. 127–62.

Brooks, R.A. (1986), ‘A robust layered control system for a mobile robot’, *IEEE Jour. of Robotics and Automation*, pp. 3–27.

Brooks, R.A. (1994), ‘Getting around Mars. Synopsis of Friday Evening Discourse’, in *Lectures April–September 1994* (London: The Royal Institution), p. 5.

Chase, W.G. and Simon, H.A. (1973), ‘Perception in chess’, *Cog. Psychol.*, **4**, pp. 55–81.

de Groot, A. (1965), *Thought and Choice in Chess*, ed. G.W. Baylor (The Hague: Mouton [English translation, with additions, of the Dutch 1946 version]).

Dennett, D.C. (1992), *Consciousness Explained* (Boston: Little, Brown & Co.).

Eysenck, H.J. and Eysenck, M. (1985), *Personality and Individual Differences* (New York: Plenum Press).

Gaines, B.R. (1995), *Inducing knowledge*. Unpublished report (Knowledge Science Institute, University of Calgary, Canada. Email: gaines@cpsc.ucalgary.ca).

Gelernter, H. (1959), 'Realization of a geometry-theorem proving machine', in *Proc. Internat. Conf. on Information Processing* (Paris: UNESCO House), pp. 273–82. Reprinted in *Computers and Thought,* ed. E.A. Feigenbaum and J. Feldman (New York, San Francisco, Toronto, London: McGraw Hill (1963), pp. 134–52).

Kaebling, L.P. and Rosenschein, S.J. (1990), 'Action and planning in embedded agents', *Robotics and Auton. Sys*., **6** (1, 2), pp. 35–48.

Kosslyn, S.M. (1980), *Image and Mind* (Cambridge, MA: Harvard University Press).

Kosslyn, S.M. (1983), *Ghosts in the Mind's Machine: Creating and Using Images in the Brain* (New York: W.W. Norton).

Maes, P. (1989), 'How to do the right thing', *Connection Science*, **1** (3), pp. 292–323.

McCarthy, J. (1959), 'Programs with common sense', in *Mechanization of Thought Processes*, Vol. 1 (London: Her Majesty's Stationery Office). Reprinted (with an added section on 'Situations, Actions and Causal Laws') in *Semantic Information Processing*, ed. M. Minsky (Cambridge, MA: MIT Press (1963)).

Michie, D. (1986), 'The superarticulacy phenomenon in the context of software manufacture', *Proc. Roy. Soc. Lond*., **A 405**, pp. 185–212, reproduced in *The Foundations of Artificial Intelligence*, ed. D. Partidge and Y. Wilks (Cambridge: Cambridge University Press (1990), pp. 411–39).

Michie (1991), 'Methodologies from machine learning in data analysis and software', *Computer Journal*, **34** (6), pp. 559–65.

Michie, D. (1995a), 'Game mastery and intelligence', in *Machine Intelligence 14*, ed. K. Furukawa, D. Michie and S. Muggleton (Oxford: Clarendon Press), in press.

Michie, D. (1995b), 'Problem decomposition and the learning of skills', in *Machine Learning: ECML-95*, Lecture Notes in Artificial Intelligence, 912, ed. N. Lavrac and S. Wrobel (Berlin, Heidelberg, New York: Springer Verlag), pp. 17–31.

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

Minsky, M. (1994), 'Will robots inherit the earth?', *Scient. Amer*., **271** (4), pp. 86–91 [the quoted passage is an abridgement from a passage in Minsky (1986)].

Nievergelt, J. (1977), 'The information content of a chess position and its implication for the chess-specific knowledge of chess players', *SIGART Newsletter*, **62**, pp. 13–15. A revised and expanded version appears in *Machine Intelligence 12*, ed. J.E. Hayes, D. Michie and E. Tyugu (Oxford: Oxford University Press, 1991).

Nilsson, N.J. (1994), 'Teleo-reactive programs for agent control', *Jour. for Art. Intell. Research*, **1**, pp. 139–58.

Nisbett, R.E. and Wilson, T.D. (1977), 'Telling more than we can know: verbal reports on mental processes', *Psych. Rev*., **84** (3), pp. 231–59.

Posner, M.I. (1973), *Cognition: an Introduction* (Glenview, IL: Scott, Foresman).

Ryle, Gilbert, (1949), *The Concept of Mind* (New York: Barnes and Noble).

Shapiro, A. (1987), *Structured Induction in Expert Systems* (Wokingham, UK, Reading, Menlo Park and New York, USA: Addison-Wesley Publishing Co.).

Shapiro, A. and Michie, D. (1986), 'A self-commenting facility for inductively synthesized endgame expertise', in *Advances in Computer Chess 4*, ed. D.F. Beal (Oxford: Pergamon), pp. 147–65.

Shapiro, A. and Niblett, T. (1982), 'Automatic induction of classification rules for a chess endgame', in *Advances in Computer Chess 3*, ed. D.F. Beal (Oxford: Pergamon), pp. 73–92.

Simon, H.A. and Gilmartin, K. (1973), 'A simulation of memory for chess positions', *Cog. Psych*., **5**, pp. 29–46.

Squire, L.R. (1987), *Memory and Brain* (Oxford: Oxford University Press).

Sternberg, M.J.E., King, R.D., Lewis, R.A. and Muggleton, S. (1994), 'Application of machine learning to structural molecular biology', *Phil. Trans. R. Soc. Lond. B*, **344**, pp. 365–71.

Trevarthen, C. (1979), 'The tasks of consciousness: how could the brain do them?', in *Brain and Mind*, Ciba Foundation Series, **69** (new series) (Amsterdam: Elsevier North-Holland).

Wilson, S. (1991), 'The animat path to AI', in *From Animals to Animats: Proc. First. Intern. Conf. on the Simulation of Adaptive Behavior*, ed. J.A. Meyer and S. Wilson (Cambridge, MA: MIT Press).