



**Birkbeck ePrints: an open access repository of the
research output of Birkbeck College**

<http://eprints.bbk.ac.uk>

Cooper, Richard (2002). Control and
communication in mental computation.
Computational Intelligence **18** (1) 29–31.

This is an author-produced version of a paper published in *Computational Intelligence* (ISSN 0824-7935). This version has been peer-reviewed but does not include the final publisher proof corrections, published layout or pagination.

All articles available through Birkbeck ePrints are protected by intellectual property law, including copyright law. Any use made of the contents should comply with the relevant law. Copyright © 2002, Blackwell Publishing Ltd. The definitive version is available at www.blackwell-synergy.com

Citation for this version:

Cooper, Richard (2002). Control and communication in mental computation.
London: Birkbeck ePrints. Available at:
<http://eprints.bbk.ac.uk/archive/00000550>

Citation for the publisher's version:

Cooper, Richard (2002). Control and communication in mental computation.
Computational Intelligence **18** (1) 29–31.

<http://eprints.bbk.ac.uk>

Contact Birkbeck ePrints at lib-eprints@bbk.ac.uk

Control and Communication in Mental Computation

Richard Cooper
R.Cooper@bbk.ac.uk
School of Psychology,
Birkbeck, University of London
Malet Street, London WC1E 7HX
United Kingdom

Frawley is correct to stress the importance of control in cognitive processing, and the essence of his proposal — that disorders of cognition can result from either within-module or between-module breakdown — is an interesting and potentially important contribution to cognitive science. However, Frawley's commitment to an overly-literal interpretation of the computational metaphor, combined with his confusion over the possible relationship between any putative language of thought and modularity in peripheral versus central processes, distracts and detracts from his work. In particular, mental processes can employ a language of thought without that language being in any way similar to standard functional or procedural programming languages. In addition, communication between (peripheral) modules does not need to be bound by a single language. Furthermore, the absence of a plausible computational system within which to demonstrate the logic/control distinction and its consequences for the language disorders under discussion weakens the within-module/between-module argument.

The computational metaphor — that the brain is like a computer — may be accepted at any one of many different levels. At one extreme, one may accept merely that the functions computed by the neural hardware are subject to the constraints of computability theory. Thus, as Frawley notes, this view is consistent even with anti-representationalist dynamic systems theory (Port & van Gelder, 1995). At the other extreme, one might conceive of the neural hardware as a glorified von Neumann machine that executes a program of instructions, and that may crash if any of those instructions embodies a false assumption (e.g., by referring to non-existent data or by attempting arithmetic operations with invalid operands). Frawley appears to opt for this second interpretation, but intermediate interpretations are possible, and arguably more desirable.

Several difficulties with Frawley's interpretation of the computational metaphor are apparent. First, it suggests that the neural hardware rigidly adheres to a predetermined instruction sequence. (Although that instruction sequence may contain conditional "if-then-else" statements, it remains a rigid instruction sequence.) Where does this sequence come from? Second, it begs the question of how an operating system complex enough to perform the necessary instruction following and allocation of cognitive and effective resources might have evolved within the neural hardware. Third, it raises the possibility of both program and operating system crashes, and the question of how the system might be rebooted. Finally, it limits our imagination with respect to non-standard operating systems and control mechanisms. In particular, it prevents us from considering operating systems and control mechanisms that are not subject to the previous three difficulties.

A limited imagination is apparent in the view, implicit in section 2.1, that all programming languages are necessarily either functional or procedural. Plausible control mechanisms that fall into neither of these categories are conceivable, and such control mechanisms have been proposed within higher cognitive domains. Thus, production systems (Newell & Simon, 1972) have proved to be highly adequate in the modelling of problem solving behaviour. Their potential in other cognitive domains has been illustrated through the development of general cognitive architectures such as Soar (Laird, Newell & Rosenbloom, 1987; Newell, 1990) and ACT-R (Anderson, 1993; Anderson & Lebiere, 1998).

Production systems do not employ linear programs in which control passes from one instruction to the next. Their behaviour is instead determined by a set of condition-action associations (corresponding to long-term knowledge) and a mechanism for selecting between competing associations when multiple

associations apply. On the assumption that the mechanism for selecting between competing associations (which is fixed at the hardware level) is robust, they cannot crash. (In the absence of additional monitoring mechanisms they can, however, lapse into infinite loops or inactivity.) Furthermore, the issues of calling subroutines (and hence whether parameters are passed by value or by address) and of banishing GOTO statements do not arise. The argument here is not that neural processing implements a production system (though it might), but that the production system model of mental computation is superior to models based on standard functional or procedural computing languages.

The rejection of standard functional or procedural computing languages for cognitive control does not imply that the issue of cognitive control cannot be informed by the structured programming movement in modern computer science. In particular, the putative functional modularity of cognitive systems shares much with the object-oriented approach to systems design (Cooper, 1995). Within this approach, complex systems are treated as collections of communicating “objects” with well defined and circumscribed functions. Objects (like modules) obey rules of information encapsulation, their operation is impenetrable, and they communicate and co-ordinate their activities through message passing. Communication and co-ordination effectively replace the standard programming concepts of logic and control. (Interestingly, one can also argue for the second tenant of the object-oriented approach — that each object is an instance or specialisation of some more generic object class. If nature were to have developed some generic mechanisms capable of computation, it would not be surprising if those generic mechanisms were not duplicated and used for a range of functions, with the different functions imposing different specialisations upon the mechanisms.)

Modularity is not at odds with a production system approach (cf. Cooper, 1995). Separate modules may effectively correspond to independent production systems (or to other computing devices). The central issue is that of communication and co-ordination between modules. This does not, in principle, require a single mentalese: any language for communication between modules may be specialised for each inter-module link (Cooper, 1994). Note also that if one accepts the modularity hypothesis as outlined by Fodor (1983), then this only concerns peripheral processes. Fodor specifically argues that central processes are non-modular. The implications are that a single language of thought is required within central processes, that peripheral processes must communicate with central processes in that language, but that central processes may use different languages to pass information to different peripheral processes.

This computationally more plausible view of the structure and function of cognitive processes does not conflict with Frawley’s analysis of SLI as effectively a within-module disorder and other linguistic disorders as between-module disorders. Rather, it presents a more convincing portrayal of Frawley’s primary distinction — that between logic and control in cognitive processes.

References:

- Anderson, J. R. (1993) *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Anderson, J. R. & Lebiere, C. (1998). *Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Cooper, R. (1994) Representation in modular networks, *PSYCHOLOQUY*, 5(88). (Commentary on R. Miikkulainen: “Subsymbolic Natural Language Processing”.
- Cooper, R. (1995) Towards an object-oriented language for cognitive modelling. In J. D. Moore and J. F. Lehman (eds.) *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, Pittsburgh, PA. pp 556–561.
- Laird, J., Newell, A., & Rosenbloom, P. (1987) Soar: An architecture for general intelligence. *Artificial Intelligence*, 33, 1–64.
- Newell, A. (1990) *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Newell, A. & Simon, H. (1972) *Human Problem Solving*. Englewood Cliffs, NJ: Prentice Hall.
- Port, R. & van Gelder, T. (eds.) (1995) *Mind as Motion: Explorations in the dynamics of cognition*. Cambridge, MA: MIT Press.