apply this explicit knowledge to improve their motor performance. That explicit knowledge can improve motor performance had been demonstrated in neurologically intact subjects[9]; Boyd and Winstein's work shows that this mechanism is available even if implicit learning has been compromised.

Boyd and Winstein's work contributes to a larger picture that is emerging in the motor skill literature. Implicit and explicit learning appear to be separable and capable of operating independently, but they can also affect one another indirectly. Explicit knowledge can guide behavior, and the behavior drives implicit learning. For example, Kelly Goedert and I asked neurologically intact subjects to perform the typical choice response-time task, but also to learn the sequence explicitly as they did so[10]. Later, we tested whether they had simultaneously learned the sequence implicitly; subjects were presented with stimuli that were mostly random, but with segments of the learned sequence occasionally slipped in. Subjects did not recognize that parts of the learned sequence occasionally appeared in this phase of the experiment, so they did not apply their explicit knowledge. Nevertheless, their response times were faster to the sequence segments than to the random stimuli, indicating that they had learned the sequence implicitly as well as explicitly.

This set of results suggests that a functional relationship exists between these two parallel systems. The explicit system can guide motor behavior, but it demands attention. As the explicit system guides behavior, the implicit system learns in the background, based on the behavior guided by the explicit system. Once the implicit system has gained sufficient knowledge the explicit system is no longer needed to guide behavior, and the skill can be executed automatically.

The Boyd and Winstein results emphasize that the explicit system can support skilled behavior when the implicit system is incapacitated, and suggests that the parietal cortex plays an important role in implicit learning. What is the neural substrate of the explicit system that can supported skilled movement? The answer to that question is not yet known. It has been suggested that the prefrontal cortex directs these explicitly driven movements. Recent neuroimaging[11] and lesion studies[12] support this conjecture, but more direct evidence is needed.

References

1 Nissen, M.J. and Bullemer, P. (1987) Attentional requirements of learning: evidence from performance measures. *Cognit. Psychol*. 19, 1–32

2 Willingham, D.B. *et al.* (1989) On the development of procedural knowledge. *J. Exp. Psychol. Learn. Mem. Cognit*. 15, 1047–1060

3 Reber, P.J. and Squire, L.R. (1998) Encapsulation of implicit and explicit memory in sequence learning. *J. Cogn. Neurosci*. 10, 248–263

4 Boyd, L.A. and Winstein, C.J. (2001) Implicit motor sequence learning in humans following unilateral stroke: the impact of practice and explicit knowledge. *Neurosci. Lett.* 298, 65–69

5 Jackson, G.M. *et al.* (1995) Serial reaction time learning and Parkinson's disease: evidence for a procedural learning deficit. *Neuropsychologia* 33, 577–593

6 Molinari, M. *et al.* (1997) Cerebellum and procedural learning: evidence from focal cerebellar lesions. *Brain* 120, 1753–1762

7 Willingham, D.B. *et al.* (2000) Implicit motor sequence learning is represented in response locations. *Mem. Cognit*. 28, 366–375

8 Honda, M. *et al.* (1998) Dynamic cortical involvement in implicit and explicit motor sequence learning. A PET study. *Brain* 121, 2159–2173

9 Curran, T. and Keele, S.W. (1993) Attentional and nonattentional forms of sequence learning. *J. Exp. Psychol. Learn. Mem. Cognit.* 19, 189–202

10 Willingham, D.B. and Goedert-Eschmann, K. (1999) The relation between implicit and explicit learning: evidence for parallel development. *Psychol. Sci*. 10, 531–534

11 Pochon, J-B. *et al.* (2001) The role of dorsolateral prefrontal cortex in the preparation of forthcoming actions: an fMRI study. *Cereb. Cortex* 11, 260–266

12 Lepage, M. and Richer, F. (2000) Frontal brain lesions affect the use of advance information during response planning. *Behav. Neurosci.* 114, 1034–1040

Daniel B. Willingham
Dept of Psychology, 102 Gilmer Hall, Box 400400, University of Virginia, Charlottesville, VA 22904, USA.
e-mail: Willingham@virginia.edu

Object-oriented models of cognitive processing

George Mather

Information-processing models of vision and cognition are inspired by procedural programming languages. Models that emphasize object-based representations are closely related to object-oriented programming languages. The concepts underlying object-oriented languages provide a theoretical framework for cognitive processing that differs markedly from that offered by procedural languages. This framework is well-suited to a system designed to deal flexibly with discrete objects and unpredictable events in the world.

Traditional approaches to computational modelling of human perceptual and cognitive processing are based on data flow through an information-processing system. The basic building blocks of such a system are information-processing modules, as depicted in Fig. 1a. Complex systems can be constructed by building sequences of these modules, the output of each module providing the input to the next, as shown in the figure. The most influential such information-processing model was proposed by Marr[1], neatly summarized by Mayhew and Frisby[2] as follows:

> '…vision is considered to be a sequence of processes that are successively extracting visual information from one representation, organizing it, and making it explicit in another representation to be used by other processes. Viewed in this way it is conceptually convenient to treat vision as computationally modular and sequential.'

This approach to modelling is clearly inspired by procedural programming languages, such as Fortran and C/C++. Programs written in these languages comprise a list of instructions to be executed in sequence in order to perform a specific task. Computational models of vision such as Marr's are often implemented in a computer program written using a procedural language. Modular information-processing models of perception and cognition can therefore be called 'procedural' models.

Over the last fifteen years a number of theories have appeared that do not conform to the procedural approach. These theories place emphasis on selective processing of perceptual 'objects'. Examples include:

- *Visual routines*[3–5]. Special self-contained processes that perform certain tasks, such as counting, indexing and tracking, and that are invoked by attention.
- *Object files*[6,7]. Representations of objects derived from early perceptual analysis that keep track of such properties as location, size and shape.
- *Fingers of instantiation* (FINST)[8]. Each object encoded from the visual scene is, according to this model, assigned a reference or pointer to it (a FINST) allowing the system to keep track of it.
- *Attention-based apparent motion*. Some forms of apparent motion emerge from attentively tracking specific perceptual objects (recently reviewed in this journal[9]).

A common thread running through these theories is the notion that visual processing is selective and flexible. It aims to construct persisting, self-contained representations of certain objects. Just as traditional processing models are closely related to procedural programming languages, object-oriented models are closely related to object-oriented programming languages, though the link has not so far been acknowledged. The first object-oriented language was Simula, developed in the mid-1960s, but wide use of such languages only began with the development of Java in the mid-1990s. An object-oriented program consists of a collection of software objects and facilities for managing those objects[10]. Each object is a self-contained set of data structures and methods (procedures) that operate on them. Objects are 'encapsulated', in the sense that their data and methods are hidden from other objects, and information is made available only by means of messages passed between objects (see Fig. 1b).

Object-oriented programs are usually event-driven, so that the functional flow of procedural programs is replaced by event processing. Execution of an object-oriented program involves an event listener or message loop that continuously listens for messages arriving from new events (such as an action from the user or from another computer, or an internal message from the computer itself), and dispatches those messages to the appropriate objects.

The principles underlying object-oriented programming languages provide a conceptual framework within which we can view object-oriented processing (OOP) models of vision and cognition, and compare them with procedural models, based on the differences between object-oriented languages and procedural languages (see Table 1).

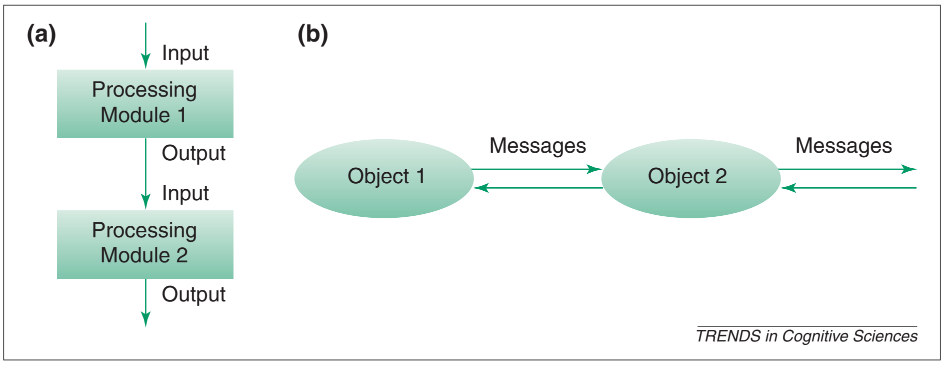

Fig. 1. Two approaches to computational models of human perceptual processing. The basic building blocks of (a) procedural processing systems, and (b) object-oriented processing systems.

Procedural models usually consist of bottom-up data flows between discrete processing modules. Processing occurs autonomously and automatically, and the representations produced are exhaustive in the sense that they attempt to build a representation of the all the visible surfaces in the scene, at least in terms of certain visual properties such as colour or motion. For example, in Marr's theory, 'a key goal of early processing is the construction of something like an orientation-and-depth map of the visible surfaces around a viewer' (Ref. 1, p. 129). Likewise Watt's theory of visual processing requires a 'full scene description'[11].

OOP models consist of encapsulated objects that communicate by means of messages. The objects may correspond to particular shapes or real-world objects in the scene, akin to 'object files'[6,7], or to encapsulated routines[3–5] that the brain can invoke as required. Processing is controlled by events (whether internally or externally generated). Shifts of attention trigger events in which new messages invoke specific routines, or instantiate new object instances. The distinction between bottom-up and top-down processing is no longer meaningful, since information flow in an object-oriented system involves bi-directional message passing between objects. Only a partial representation of the scene is constructed, based on some of the objects present in the scene and their relations. In order to interact with the world we do not need a complete scene description, but a way of encoding and tracking objects in it[12]. Objects are selected for encoding by attention-attracting 'events'.

Evidence consistent with OOP models includes:

(1) Object-specific priming[7]: priming effects in letter-naming tasks are specific to the visual object on which the letter is displayed, consistent with the notion of object files.

(2) Multiple object tracking[8]: observers can keep track of no more than four randomly moving objects at a time, indicating that the visual system has a way of selecting and accessing only a small number of objects at a time.

(3) Inattentional blindness[13,14]: observers are unable to detect even gross changes in image content when they occur during saccades or simulated saccades, indicating that much less scene information is encoded with each glance than previously thought.

(4) Attentional effects in motion perception[9]: the perceived direction of image motion and the strength of motion adaptation are both amenable to modulation by attention, contrary to views that low-level motion processing is autonomous.

Some aspects of cortical physiology are also consistent with OOP. The multitude of reciprocal connections between different visual areas in the cortex[15] seems untidy from the point of view of data flow between modules, but less mysterious from the point

Table 1. Comparison of procedural and object-oriented processing (OOP) models

	Procedural models	OOP models
Architecture	Modular	Encapsulated
Process	Data flow	Message passing
Control	Autonomous & sequential	Event driven
Representation	Exhaustive	Partial

of view of bi-directional message passing between processing objects. In an extensive review of parallel processing streams, Lennie argues that the fundamental job of the visual system is to 'classify objects quickly and reliably'[16]. He takes issue with the modular processing model of cortical organization, according to which information streams through a series of processing stages, arguing instead that:

'Each level [of cortical processing] passes on to the one above only the results of its analysis, so that, for example, the local continuity that V2 finds is expressed to V4 as a continuous structure of a certain class, but V4 knows nothing about the information used to discover that structure…each area passes on *decisions* about image structure at a particular level of analysis…' (Ref. 16, p. 918)

Lennie therefore favours an encapsulated representation characteristic of OOP systems. Encapsulation means that a specific object hides its data from other objects, and communicates by means of messages instead (*decisions* in the quote above).

Simula, the first object-oriented programming language, was designed for discrete event simulation. As the world consists of complex objects that change state and influence events unpredictably, it was natural to simulate it using messages that pass between objects and cause changes in object state. In this way, the structure of the programs reflected the structure of the problem[10]. Many researchers believe that the goal of the brain is to build some kind of internal model of the world, an idea dating back to Kenneth Craik in the 1940s[17]. Just as in the case of simulation modelling, it would be beneficial for the brain's computational structure to reflect the structure of the world. A gradual realization of this fact could lie behind the growth of OOP models of perception and cognition. An explicit acknowledgement of the conceptual link between OOP models and object-oriented languages might promote the development of more sophisticated OOP models.

References

1 Marr, D. (1980) *Vision*, Freeman

2 Mayhew, J. and Frisby, J. (1984) Computer vision. In *Artificial Intelligence: Tools, Techniques, and Applications* (O'Shea, T. Eisenstadt, M., eds), pp. 301–357, Harper & Row

3 Thornton, I. *et al.* (2000) The role of attention in the processing of biological motion. *Perception* 29, A114

4 Cavanagh, P. *et al.* Attention-based visual routines: sprites. *Cognition* (in press)

5 Ullman, S. (1984) Visual routines. *Cognition* 18, 97–159

6 Kahneman, D. and Treisman, A. (1984) Changing views of attention and automaticity. In *Varieties of Attention* (Parauraman, R. and Davies, D., eds), pp. 29–62, Academic Press

7 Kahneman, D. *et al.* (1992) The reviewing of object files: object-specific integration of information. *Cognit. Psychol.* 24, 175–219

8 Pylyshyn, Z. and Storm, R. (1988) Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spat. Vis.* 3, 1–19l

9 Raymond, J. (2000) Attentional modulation of visual motion perception. *Trends Cognit. Sci.* 4, 42–50

10 Graham, I. (1994) *Object Oriented Methods*, Addison-Wesley

11 Watt, R. (1988) *Visual Processing*, Erlbaum

12 Pylyshyn, Z. (2000) Situating vision in the world. *Trends Cognit. Sci.* 4, 197–207

13 Simons, D.J. and Levin, D.T. (1997) Change blindness. *Trends Cognit. Sci.* 1, 261–267

14 Simons, D.J. (2000) Attentional capture and inattentional blindness. *Trends Cognit. Sci.* 4, 147–155

15 Van Essen, D. *et al.* (1992) Information processing in the primate visual system: an integrated systems perspective. *Science* 255, 419–423

16 Lennie, P. (1998) Single units and visual cortical organization. *Perception* 27, 889–935

17 Craik, K. (1943) *The Nature of Explanation*, Cambridge University Press

George Mather
Dept of Experimental Psychology, Biology School, University of Sussex, Brighton, UK BN1 9QG.
e-mail: georgem@biols.susx.ac.uk

Asymmetries in preparation for action

John L. Bradshaw

The origins and nature of hemispheric specialization of action control are unclear. A review of some recent evidence suggests that the right hemisphere interprets spatial relationships whereas the left deals with temporal control of movement. Contrary to the popular view, specialization of the right hemisphere for spatial representations might have preceded left hemisphere specialization for language and movement.

Most of us are right handed, and will suffer a more-immediately-catastrophic loss of speech and language function after left-sided brain damage. Such damage will also impair our 'praxic' skills at meaningfully, correctly and appropriately manipulating tools or instruments, or performing complex gestures. Note that apraxia may be defined as impaired execution of learned, skilful, meaningful or purposeful actions, by *either* hand, when accurate spatio-temporal patterning is important. A range of deficits have been invoked in apraxia, including loss of visuokinesthetic motor representations (posterior parietal), a disruption to the kinematics of movement (frontoparietal?), a loss of knowledge of the meaning of movement (temporoparietal?), an impaired ability to relate movement to sensory cues (lateral premotor?), or, especially, to properly *sequence* movement (supplementary motor area?).

It is as yet unresolved whether the evolution and lateralization of language and praxic functions were and are interdependent, and how unique we are in these respects[1]. Is sequencing the common link between language and praxis, and, if so, how does sequencing relate to known left-hemisphere functions? Why do we have cerebral asymmetry? Two recent studies from the same Oxford group re-open these issues.

Apraxia

Rushworth *et al.*[2] studied a group of apraxic and non-apraxic patients, and concluded that although apraxia resulting from left hemisphere injury often affects the capacity to sequence movements, nevertheless some forms of impairment affect response selection rather than sequencing *per se*.

This year, Schluter *et al.*, in a complementary study[3], used PET to investigate cerebral dominance for the selection of action. In one condition, normal, healthy subjects moved one of two fingers depending on the cue presented (choice reaction time), and in another they moved the same finger whatever the cue (simple reaction time). Activations occurred in prefrontal, premotor and intraparietal

PII: S1364-6613(00)01656-9