Hidden-Object Indexing: Claims at Two Temporal Levels

Erik M. Altmann
Krasnow Institute for Advanced Study
George Mason University
Fairfax, VA 22030
altmann@gmu.edu

This abstract is concerned with the memory phenomena underlying directed access to hidden external information. Such access depends on knowing what information exists in the environment. That is, it depends on having a memory for hidden objects that functions as an *index* in which to look them up when they are relevant to the task at hand.

Examples of hidden-object indexing come from a study in which we observed a programmer working in a real-world task environment (Altmann, 1996; Altmann, Larkin & John, 1995). The programmer interacted with an interpreted program. She issued multiple queries and run commands to the interpreter, thereby generating a large buffer of output. Only a fraction of this buffer could fit on the screen at once, but hidden portions could be scrolled back into view. Occasionally the programmer scrolled back to some hidden object. These scrolling events, as well as the original appearance of each scrolled-to object, were captured in a log of process data that included verbal and keystroke protocols and all screen changes.

Altmann (1996) describes five of these scrolling events in detail. In two of them, the object was hidden for less than 30 seconds, and during that time remained closely relevant to the programmer's current goal. For example, in one case the programmer wanted to compare two program fragments, with one fragment hidden. The programmer probably kept the hidden fragment active in working memory (WM) as a basis for comparing to the visible fragment.

In the three remaining scrolling events, the scrolled-to object was hidden for up to several minutes, during which the programmer performed other tasks. This implicates long-term memory (LTM), because the object is unlikely to have remained active in WM the entire time it was hidden. Apparently some trace was stored in LTM when the object was first attended, and was retrieved later as a basis for deciding to scroll back to the object. The protocol data show no evidence that the trace was stored prospectively out of anticipation for future recall demands; the encoding seems to have been largely incidental.

We implemented a computational model spanning all five scrolling events and intervening behavior (Altmann, 1996). For each event the model encodes the object of interest when it first appears on the screen. Later, the model scrolls to that object based in part on the trace encoded earlier. The model is constrained on two fronts: the underlying architecture (Soar; Newell, 1990) imposes theoretical constraints on learning and memory, and the behavioral data impose different functional requirements on the memory for an object, depending on whether the object was hidden recently or not.

In the short term, the model must be able to maintain an active record of an object's existence while that object is hidden. This is implemented as follows. If an object encoded in WM disappears from the screen, the model tags the corresponding WM code as *recently-hidden*. The [object, recently-hidden] pair persists briefly in WM after the object becomes hidden, for an interval spanning at least the current goal. During this interval the model remains aware of the hidden object. The recently-hidden tag is cheap to compute, because the visual process that monitors the screen exploits high-level spatial knowledge that prevents having to monitor every object. (The model must have this spatial knowledge anyway, to account for an independent dimension of the programmer's behavior.) Thus the theoretical claim is that people maintain an active index of recently-hidden objects, and that this maintenance is cheap enough to be automatic.

In the long term, the model must store some record of an object in LTM, and be able to activate this record later should the object become relevant. The model encodes *episodic* traces in LTM, as a side effect of attention. Each such trace represents the event of attending to an object. The trace is associative, in that it requires a cue for retrieval. The required cue is an image of the attended object appearing in WM. When the object is hidden, this image must be generated from memory. Thus the model must generate images (from LTM) in order to retrieve episodic traces (also from LTM). This is too costly to be plausible for short-term indexing, because it involves multiple successive retrievals from LTM. However, encoding is cheap enough to be automatic, as a result of Soar's learning mechanism. Thus the claim is that people automatically acquire pointers to objects as a side effect of attention, and activate these pointers when necessary using a deliberate, knowledge-based process of cue generation.

References

Altmann, E.M. (1996). *Episodic Memory for External Information.* Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Altmann, E.M., Larkin, J.H. & John, B.E. (1995). Display navigation by an expert programmer: A preliminary model of memory. *CHI 95 Conference Proceedings.* New York: ACM Press.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge: Harvard University Press.