HUMAN-COMPUTER INTERACTION, 1997, Volume 12, pp. 301-309

Introduction to This Special Issue on Cognitive Architectures and Human-Computer Interaction

Wayne D. Gray
George Mason University

Richard M. Young
University of Hertfordshire

Susan S. Kirschenbaum
Naval Undersea Warfare Center Division, Newport, RI

This special issue was assembled by editors and contributors who believe that cognitive architectures provide the most important new contribution to a theoretical basis for HCI (human-computer interaction) since the publication of *The Psychology of Human-Computer Interaction* (Card, Moran, & Newell, 1983). In this introduction, we provide a brief overview of what cognitive architectures are and why we find them exciting. Then we introduce the four architectures represented by articles in this special issue.

Wayne D. Gray is a cognitive scientist with an interest in how artifact design affects the cognition required to perform tasks; he is Chair of the Human Factors and Applied Cognition Program at George Mason University. **Richard M. Young** works in the Department of Psychology at the University of Hertfordshire; he is interested in cognitive modeling and its application to human-computer interaction. **Susan S. Kirschenbaum** is an an engineering psychologist in the Department of Combat Systems at the Naval Undersea Warfare Center Division, Newport, RI, where she studies and models expert submarine decision-makers.

WHAT IS A COGNITIVE ARCHITECTURE?

According to Howes and Young (in this special issue), "A cognitive architecture embodies a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task." A sufficiently complete cognitive architecture will ensure that theories of, say, reading and comprehension rely on the same mechanisms of information processing, storage, and retrieval that have been shown to be valid for playing chess, handling air traffic control, doing transcription typing, using a hand calculator, and programming a VCR. Rather than postulating a different set of mechanisms and parameters for each phenomenon studied, a cognitive architecture attempts to apply the same core set of constructs across an entire range of phenomena. Currently, all cognitive architectures remain research projects. All are incomplete.

Cognitive architectures differ from traditional cognitive theorizing in three important ways. First, traditional cognitive theories focus on building microtheories of isolated phenomena or mechanisms (Newell, 1973), whereas cognitive architectures seek integration. Second, attempts to show how theoretically distinct mechanisms combine, interleave, or contribute to complex cognition lead many designers of cognitive architecture to emphasize the control structure of cognition. Third, and perhaps most relevant for HCI, this emphasis on control structure has led most cognitive architectures to be cast in a form amenable to computer implementation, so that individual models built within the architecture take the form of computational cognitive models.

Constructing a model within one of these cognitive architectures involves writing a program in a powerful, restricted, and specialized programming language. The languages must be powerful if they are to enable the modeler (i.e., the programmer) to build models that can emulate even a fraction of the tasks performed by human cognition. The languages are restricted. Whereas traditional programming languages are designed to give the programmer great flexibility in implementing an algorithm, the point of a computational cognitive model is that the data structures and algorithms used are constrained by cognitive theory. We are not talking about artificial intelligence programming, in which the goal is, for example, to write a computer program that will beat Garry Kasparov at chess. This is computational cognitive modeling, in which the goal is to write a computer program that will play chess the same way Garry Kasparov does. The former has been done; the latter remains a significant challenge. Finally, the programming languages of cognitive architectures are specialized insofar as each was developed to emphasize certain aspects of human cognition. Although the goal may be a unified theory of all human cognition, there are many paths leading toward that goal, and, from different starting points, each current cognitive architecture follows a different route.

WHY ARE COGNITIVE ARCHITECTURES IMPORTANT FOR HCI?

Cognitive architectures are in the process of becoming the preferred route for bringing cognitive theory to bear on HCI. Several of their strengths provide help for meeting the challenge of understanding, evaluating, and engineering the interactions between humans and their information artifacts:

- Cognitive architectures provide the mechanisms and processes by which different aspects of cognition—attention, memory, problem solving, learning, and so forth—work in concert to exhibit integrated behavior of the kind required to perform tasks with interactive computer interfaces.
- This degree of integration provides a starting point for analyzing (and then resynthesizing) the cognition that occurs at the interface. When confronted with an interface, the cognitive modeler asks, "How would C–I [or EPIC, Soar, ACT–R, etc.] perform the task using this design?" The answer is cast in terms of a complete theory (a working model) of all the cognition required to do the task using the interface in question. In contrast, traditional methods of cognitive theorizing tend to be phenomenon based (Newell, 1973). Narrow-scope theories of divided attention or decision making—or whatever—focus on those elements of the interaction that best illustrate their defining phenomena, whereas other elements tend to be ignored.
- Computational models derived from cognitive architectures can be applied to tasks involving both routine skill and problem solving. Skilled cognition is contrasted to that requiring problem solving, with the distinction involving the amount of control knowledge used to guide search through a problem space (Card et al., 1983). Whether a particular task is routine will of course depend on the knowledge and experience of the individual, but typical examples of skilled tasks include transcription typing (for a skilled typist) and copying files (for a skilled computer user). Examples of tasks involving a large component of problem solving include computer programming, scientific reasoning, and most design activities. Many tasks involve components of both skilled cognition and problem solving. Unlike most other approaches to cognitive modeling, cognitive architectures can handle tasks across the continuum.
- Computational cognitive models based on cognitive architectures are generative in that the same model can be applied to a range of different tasks or variations on a task. This contrasts with more descriptive analyses, for which new tasks, or variations in performing an already modeled task, require the development of new descriptive models.

- Finally, there is the issue of learning. Although static computational models can be used to measure what must be learned and to predict learning time (e.g., Bovair, Kieras, & Polson, 1990), only models based on cognitive architectures can model the learning process itself. Hence, architecture-derived computational models can be used to determine how different tasks or instructions lead to different types and amounts of learning and how forgetting or other causes of memory failure underlie the difficulties encountered by casual users of complex systems.

Besides providing a conduit for applying theory to design, cognitive architectures are also on the path driving the development of cognitive theory from applied problems in HCI. Interacting with modern computer systems requires the user to respond to text and graphic events as they occur. It also requires a complex orchestration of cognition with perception and motor processes. This challenge to cognitive theory to account for the event-driven, display-based performance required by graphical user interfaces (GUIs) and multimedia interactions has brought renewed attention to HCI phenomena from the cognitive science community. These researchers are looking at our field with new eyes and are finding that minor differences in how an interface is designed can lead to significant differences in the interplay among cognition, perception, and motor activities and thence to large differences in performance. For example, studies of decision making have shown that the cognitive cost of processing (Kleinmuntz & Schkade, 1993; Payne, Bettman, & Johnson, 1993) or accessing information (e.g., by eye movements vs. moving a mouse to its location; Lohse & Johnson, 1996) leads the user to adopt different decision-making strategies that vary in their effectiveness.

In the short run, this influx of basic researchers into HCI means that most articles written about the application of cognitive architectures will be of more interest to researchers than to practitioners. In the long run, however, practitioners will benefit. Driven by the needs of the research community, HCI will become the first applied field in which phenomena can be fully accounted for by cognitive architectures. Furthermore, the skills possessed by the HCI community are exactly those required to take such research tools and transform them into tools for practitioners. The day of the routine use of cognitive architectures to model HCI tasks may not yet be upon us, but it is not far away.

THE ARCHITECTURES AND ARTICLES IN THIS SPECIAL ISSUE[1]

Until about the mid-1980s, it was possible to identify only one candidate for a cognitive architecture—ACT* (Anderson, 1983). Today, we can list about half a dozen. Of the cognitive architectures that have emerged since 1983, Soar (Newell, 1990) is the most established and best known within the HCI community. Unlike the ACT family, Soar has always had a programming language as an integral part of the theory. This has meant that, since its inception, to theorize in Soar has entailed writing a computational cognitive model to perform the behavior in question. One criterion for a successful theory is its acceptance by researchers outside its circle of developers. On this basis, Soar clearly leads the pack, as well-established communities of Soar modelers exist throughout the United States and Europe.

In this special issue, Howes and Young use Soar as a springboard to explore how an architecture constrains the way that models of behavior must be constructed. Howes and Young illustrate their points with the seemingly roundabout way that Soar models learn to associate menu labels with actions—that is, how Soar learns to use a GUI. An interesting and important conclusion is that, although the natural way to implement methods for this task in Soar seem very indirect from a traditional programming perspective, the models appear to capture the essence of human performance.

The construction–integration (C–I) architecture started life as a theory of language comprehension (Kintsch, 1988). Developments by Mannes and Kintsch (1991) as well as Doane, Pellegrino, and Klatsky (1990) transformed C–I into a tool for modeling HCI tasks. In this special issue, Kitajima and Polson present the latest installment of their LICAI (linked model of comprehension-based action planning and instruction taking) extensions to C–I theory. As with previous versions (Kitajima, 1989; Kitajima & Polson, 1992, 1995, 1996), the current article focuses on the cognition involved when an experienced GUI user attempts to learn a new software package. An interesting contribution of the article is its development and elaboration of mechanisms for goal selection within a C–I architecture.

EPIC (executive process–interactive control) was developed by Meyer and Kieras (1997a, 1997b) for the stated purpose of representing embodied cognition. Compared to other cognitive architectures, EPIC is an idiot

1. Further cognitive architectures that are relevant to HCI but that are not represented in this special issue include CAPS (e.g., Byrne & Bovair, 1997), ZIPPY (e.g., Rist, 1995), and ICS (e.g., Barnard & May, 1993; May, Barnard, & Blandford, 1993).

savant: Its cognitive component is minimized in favor of well-developed motor and perceptual components. This intentionally lopsided development permits EPIC modelers to determine constraints on cognitive performance that emerge from properties of the human's input (auditory and visual perception) and output (speech and motor movement) mechanisms. A modeler using EPIC must adhere to the parameters provided for moving eyes, processing vision, and so forth. In this special issue, Kieras and Meyer present HCI researchers with a detailed introduction to EPIC. In addition, they present data and model predictions for several tasks of interest to HCI researchers, including a menu search task originally studied by Nilsen (1991; see also Hornof & Kieras, 1997).

The original ACT* architecture (Anderson, 1983) metamorphized first into ACT–R (Anderson, 1993) and now into ACT–R 4.0 (Anderson, in press). Along the way, it has expanded in coverage, acquired a fully implemented programming language, and grown eyes and fingers. Indeed, in this special issue, Anderson, Matessa, and Lebiere argue that the newly augmented ACT–R is now fully capable of handling the cognition that occurs at the computer interface. Their claim is that ACT–R now has a theory of visual perception (discussed in detail elsewhere) and visual attention (the focus of their current article). These two components expand ACT–R's coverage so that it can now be used to model human interaction with computer applications. The validity of the claim is demonstrated by the application of ACT–R to several classic but never before modeled psychological paradigms.

One of the tasks modeled by Anderson, Matessa, and Lebiere is the menu search task (Nilsen, 1991) modeled by Kieras and Meyer. After developing an ACT–R model, Anderson et al. apply it to a new menu search task. They claim that ACT–R predicts human performance in this new task using the same parameters and assumptions required to model the first task. They further claim that the model developed by Kieras and Meyer would need additional assumptions to fit the new data. This comparison provides an interesting example of two different architectures tackling the same benchmark task. Such competitive argumentation (VanLehn, Brown, & Greeno, 1982) or cooperative analysis across shared scenarios (Young & Barnard, 1987) may offer the best ways to compare the relative merits of different cognitive architectures.

CAVEATS AND CONCLUSIONS

In this introduction, we have argued for the importance of cognitive architectures as a vehicle for injecting cognitive theory into HCI. However, the critical importance of cognitive architectures is far from universally accepted, and we do not expect to convince those who believe that cognitive theory has no part to play in HCI. Moreover, despite our espousal of cognitive architectures for HCI, we must acknowledge two important caveats.

First, by focusing on cognitive theory, we do not mean to imply that other behavioral and social sciences have no role in HCI. For example, the research community on computer-supported cooperative work (CSCW) brings a variety of cognitive and noncognitive sciences to bear on HCI issues. Some of these issues are outside the scope of cognitive theories. In addition, some of the cognitive issues addressed are closely intertwined with social issues in ways that are not currently amenable to architecture-based approaches. Second, there are basic cognitive processes that are important to HCI but that are not sufficiently well understood to be incorporated into existing cognitive architectures. Clear examples of such areas come from speech understanding and visual perception.

Although these caveats temper our claims, they do not temper our enthusiasm. Despite a gestation period that began in the 1950s (Simon, 1991, chap. 12), the advent of runnable cognitive architectures is a relatively recent occurrence. For example, between 1983 and 1993, 11 articles either foreshadowed or applied something like an architecture-based approach to HCI phenomena. In contrast, since 1993, 21 articles (including the 4 in this special issue) have built architecturally inspired models of HCI tasks. We see cognitive architectures as a new phenomenon—one that is gaining wide acceptance within the cognitive science community and whose potential for application to HCI concerns is just beginning to be appreciated.

NOTES

Background. The articles in this special issue have their origins in a workshop conducted at the CHI'95 Conference on Human Factors in Computing Systems, Denver, CO (Kirschenbaum, Gray, & Young, 1996).

Support. Wayne D. Gray's work on this special issue was supported in part by Office of Naval Research Grant N00014-95-1-0175 and a Fellowship from the Krasnow Institute for Advanced Studies. Susan S. Kirschenbaum's work on this special issue was supported in part by the Office of Naval Research (Project A10328) and the Naval Undersea Warfare Center's Independent Research Program (Project E10207).

Editors' Present Addresses. Wayne D. Gray, George Mason University, MSN 3f5, Fairfax, VA 22030. E-mail: gray@gmu.edu. Richard M. Young, Department of Psychology, University of Hertfordshire, Hatfield, Herts, AL10 9AB, United Kingdom. E-mail: r.m.young@herts.ac.uk. Susan S. Kirschenbaum, Code 2214, Building 1171/1, Naval Undersea Warfare Center Division, Newport, RI 02841. E-mail: kirsch@c223.npt.nuwc.navy.mil.

REFERENCES

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, J. R. (in press). *Atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Barnard, P. J., & May, J. (1993). Cognitive modeling for user requirements. In P. Byerley, P. J. Barnard, & J. May (Eds.), *Computers, communication and usability: Design issues, research and methods for integrated services* (pp. 101–146). Amsterdam: North-Holland.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human–Computer Interaction, 5,* 1–48.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science, 21,* 31–61.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human–computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Doane, S. M., Pellegrino, J. W., & Klatsky, R. L. (1990). Expertise in a computer operating system: Conceptualization and performance. *Human–Computer Interaction, 5,* 267–304.

Hornof, A. J., & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems,* 107–114. New York: ACM.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction–integration model. *Psychological Review, 95,* 163–182.

Kirschenbaum, S. S., Gray, W. D., & Young, R. M. (1996). Cognitive architectures and HCI [Workshop report]. *SIGCHI Bulletin, 28*(2), 18–21.

Kitajima, M. (1989). A formal representation system for the human–computer interaction process. *International Journal of Man–Machine Studies, 30,* 669–696.

Kitajima, M., & Polson, P. G. (1992). A computational model of skilled use of a graphical user interface. *Proceedings of the CHI'92 Conference on Human Factors in Computing Systems,* 241–249. New York: ACM.

Kitajima, M., & Polson, P. G. (1995). A comprehension-based model of correct performance and errors in skilled, display-based, human–computer interaction. *International Journal of Human–Computer Studies, 43,* 65–99.

Kitajima, M., & Polson, P. G. (1996). A comprehension-based model of exploration. *Proceedings of the CHI'96 Conference on Human Factors in Computing Systems,* 324–331. New York: ACM.

Kleinmuntz, D. N., & Schkade, D. A. (1993). Information displays and decision processes. *Psychological Science, 4,* 221–227.

Lohse, G. L., & Johnson, E. J. (1996). A comparison of two process tracing methods for choice tasks. *Organizational Behavior and Human Decision Processes, 68*(1), 28–43.

Mannes, S., & Kintsch, W. (1991). Routine computing task: Planning as understanding. *Cognitive Science, 15,* 305–342.

May, J., Barnard, P. J., & Blandford, A. E. (1993). Using structural descriptions of interfaces to automate the modeling of user cognition. *User Modeling and User-Adapted Interaction, 3,* 27–64.

Meyer, D. E., & Kieras, D. E. (1997a). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review, 104,* 3–65.

Meyer, D. E., & Kieras, D. E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory-period phenomena. *Psychological Review, 104,* 749–791.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual information processing* (pp. 283–308). New York: Academic.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Nilsen, E. L. (1991). *Perceptual-motor control in human–computer interaction* (Technical Report 37). Ann Arbor: University of Michigan, Cognitive Science and Machine Intelligence Laboratory.

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1993). *The adaptive decision maker.* New York: Cambridge University Press.

Rist, R. S. (1995). Program structure and design. *Cognitive Science, 19,* 507–562.

Simon, H. A. (1991). *Models of my life.* New York: Basic.

VanLehn, K., Brown, J. S., & Greeno, J. (1982). Competitive argumentation in computational theories of cognition. In W. Kintsch, J. Miller, & P. Polson (Eds.), *Methods and tactics in cognitive science.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Young, R. M., & Barnard, P. (1987). The use of scenarios in human–computer interaction research: Turbocharging the tortoise of cumulative science. *Proceedings of the CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface,* 291–296. New York: ACM.

ARTICLES IN THIS SPECIAL ISSUE

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT–R: A theory of higher level cognition and its relation to visual attention. *Human–Computer Interaction, 12,* 311–343.

Howes, A., & Young, R. M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human–Computer Interaction, 12,* 345–389.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human–computer interaction. *Human–Computer Interaction, 12,* 391–438.

Kitajima, M., & Polson, P. G. (1997). A comprehension-based model of exploration. *Human–Computer Interaction, 12,* 439–462.