“Timelines” provides perspectives on HCI history, glancing back at a road that sometimes took unexpected branches and turns. History is not a dry list of events; it is about points of view and differing interpretations.

Jonathan Grudin, Editor | jgrudin@microsoft.com

Multiscale Zooming Interfaces: A Brief Personal Perspective on the Design of Cognitively Convivial Interaction

James D. Hollan
University of California, San Diego | hollan@cogsci.ucsd.edu

Editor’s note: I first worked with Jim Hollan as a grad student. I was a teaching assistant in the cognitive psychology laboratory course at UCSD 31 years ago. I worked for him again in the MCC Human Interface Laboratory (see Bill Curtis’s November + December 2010 “Timelines” column for his account of MCC). Every conversation I’ve had with Jim since has been fun and rewarding. He is one of the most thoughtful HCI researchers. From his work on Steamer, his co-authorship of a 1986 chapter on direct manipulation that is still worth reading, the influential “beyond being there” paper, to subsequent work on zooming interfaces, he has been at the forefront of the fields of HCI and design.

Ideas are like people, in that the more we know about their history, the better we can appreciate them. Although we will never know the full history of even a simple idea, the more we know the richer and more nuanced our understanding becomes [1]. In this short account I provide a highly selective and personal history of the development of zooming or multiscale interfaces. The main point I want to make is that although a part of what makes zooming appealing is how naturally it fits with our everyday experience of approaching objects to see them in greater detail, a much more important perspective for design is to view zooming as an instance of a cognitively convivial physics of interaction that can not only exploit perceptual and cognitive abilities but also match multiple semantic levels of tasks. Semantic zooming, for example, allows movement along conceptual rather than physical dimensions. But I am getting ahead of the story.

When I formed the Computer Graphics and Interactive Media research group at Bellcore in the early 1990s, I created an opportunity to further explore what I thought of as cognitively inspired interfaces—interfaces designed to exploit what we understand about cognition. This approach was derived from my earlier experiences designing a series of dynamic interactive graphical systems: Steamer—a system to aid people in understanding complex dynamic systems [2], Moboard—a system to assist in learning to solve navigational problems [3], and HITS (Human Interface Tool Suite) [4], built by the Human Interface Lab at MCC that I led. It was also motivated by work with Ed Hutchins and Don Norman [5], in which we attempted to provide a cognitive account of both the benefits and weakness of direct manipulation, in reaction to Shneiderman’s early characterization of direct manipulation interfaces [6].

Central to these systems were domain-specific editors to ease interface design and implementation. The goal was to enable designers to contribute to interface development at a level and focus appropriate to their specific expertise. At higher levels, designers manipulated graphical representations, and the editor created the associated code. For example, Steamer was built around a graphical editor that allowed subject-matter experts with no programming experience to assemble dynamic interactive views of a complex simulation at multiple levels of detail. The objective was to create graphical instantiations of the qualitative models that experts employed in their explanations of system operation. Our design mantra was “conceptual fidelity rather

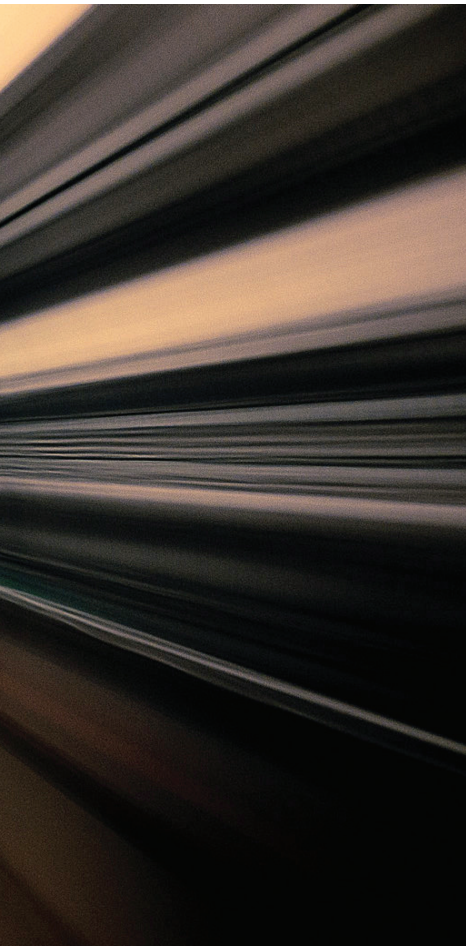
Photograph by Sam Javanrouh

than physical fidelity." The Steamer graphical editor became the basis for the HITS system, which was expanded to include additional editors to support multimodal interfaces (graphical, natural language, gestural, and sketch-based) for what we termed high-functionality systems. HITS was built on an underlying knowledge base and provided a run-time environment to support multimodal interaction.

One deep idea in HITS was the notion of creating a "tool chain" to permit dynamic changes to be made at multiple levels. This was enabled by an integrated representation of the interface and, in fact, of the editors themselves, that allowed modifications from low-level details to high-level, task-specific characteristics. This multilevel software development environment was designed to encourage interface evolution and integration of multiple modalities. We thought it was a fundamentally flawed notion that interface design had to be accomplished either at a low level by skilled programmers or via high-level tool kits that, while requiring less programming expertise, highly constrained the range of what could be designed. We were convinced that to fully exploit the expertise of designers and encourage evolution of an interface over time required a tool chain that connected and integrated the multiple levels of design necessary to span the enormous distance between the low-level bit-shuffling of machines and the complex situated tasks of users.

At the time I started the group at Bellcore, it was becoming clear that the graphical hardware that was required to enable the kinds of dynamic graphical interfaces that we had been exploring in Steamer and HITS would move from machines that cost in the six digits to inexpensive boards that cost hundreds of dollars and would become commonplace on personal computers. We and others were excited by the radical changes that such commonly available graphical facilities would enable. At around the same time, Stu Card at PARC was forming a new group to explore 3-D interfaces and information visualization. Stu's group built its seminal Information Visualizer system [7]. At Bellcore, we too were interested in dynamic 3-D interfaces for information visualization and were building a system we called ART3 to explore information visualization of complex data [8].

Another notion that motivated our effort was the conjecture that too much of interface design involved replication of earlier media. The central notion of the "desktop metaphor" was an interface that



mimicked the physical desktop of documents and folders. While there is certainly considerable power in exploiting what people already know and are used to, it seemed unlikely to fully exploit the power of computation. In those days, when I gave talks, I used a slide from Don Gentner and Jonathan Grudin that showed a steam-driven tractor. The interface for controlling this vehicle was a set of reins. One can understand how logical it was to adopt this commonly understood interface at a time of movement from horse power to steam power. Our view was that much of then current interface work was likely to soon be seen as being just as anachronistic. More important, following a replication approach highly constrained the space of possible interfaces. I started to think of the design of interfaces as being the design of cognitively inspired physics of interaction to facilitate and simplify cognitive tasks rather than just mimicking the physics of older media. One simple but influential example for me was snap-dragging, in which the mouse pointer snaps to points on the grid. This is a physics that nicely supports grid-based graphical layout. Andy Witkins's wonderful early work on interactive dynamics was another influence [9].

As we were developing ART, Ken Perlin from NYU visited and demonstrated Pad, an early zooming interface developed by Ken and his student David Fox [10]. This prototype provided an exciting view of a radical multiscale interface alternative. For me it reinvoked the myriad benefits of the multiscale interfaces we had built in Steamer but was different because of providing zooming rather than discrete shifts between levels. It was an example of a dynamic interactive physics like we were developing, but instead of providing a 3-D interface, it was a 2-D interface in which any object could be placed at any place and at any scale. We quickly began to implement and explore this in the system we were building by making any polygon in our 3-D world function as a zoomable 2-D surface. At around this time, Ben Bederson joined my group, and he and I both became increasingly less interested in exploring 3-D interfaces and more and more interested in zoomable 2-D interfaces. Part of this was the appealing simplicity of navigation in 2-D versus 3-D worlds, and another was the rich space of interface possibilities we saw in 2-D zoomable worlds [11].

Also around this time, I left Bellcore to become chair of the computer science department at the University of New Mexico; Ben shortly thereafter joined me as a young faculty member. We teamed up with Ken Perlin and George Furnas—who had taken over my group at Bellcore and were responsible for seminal early work at Bell Labs on fish-eye lenses for interacting with information [12]—and I put together a proposal to DARPA to create a multiscale interface environment we called Pad++. We received very generous funding to support the development of Pad++ [13, 14] and made the system widely available to the research community. We like to think that Pad++ helped to spawn the current widespread adoption of zoomable interfaces.

Reflecting back on this early development effort, a decade and a half after Pad++, I clearly see that the intellectual lineage of the work extends back to Sutherland's seminal Sketchpad system from the 1960s (What doesn't?), but it was early experiences I had at Xerox PARC as a consultant for John Seely Brown's Cognitive and Instructional Sciences group in the late 1970s and early 1980s that was personally most influential. The research environment at PARC was a tremendous influence on my views of interfaces and software tools. These were the early days of Smalltalk and beginnings of personal computing: the amazing Alto and subsequently personal Lisp Machine (the Xerox D-machines—Dolphin, Dandelion, and Dorado—and the Lisp Machine from MIT and subsequently Symbolics). Access to an Alto, and later to both Xerox and Symbolics Lisp Machines, changed and motivated the direction of my research and interest in exploring cognitively inspired physics for interfaces.

I would now argue there is an important impedance match between zoomable, multiscale interfaces and human cognition. Because of the nature of vision and our experience with the world, it is natural to move closer to an object to see it in more detail and to move away to see the larger context. It is important to realize that while geometric scaling is natural and provides multiple cognitive benefits, e.g., helping to maintain object permanence as one moves in a space, computationally based forms of interaction can be designed that provide the ability to interact directly with semantically meaningful aspects of tasks. If one focuses only on mimicking the physics of the world, one isn't led to consider how to improve on such physics nor to develop physics for interacting with conceptual aspects of domains. Just as snap dragging facilitates the task of grid layout, semantic zooming goes beyond simple geometric zooming to allow navigation in the multiple semantic coordinate systems of meaningful tasks. Rather than just changes in geometric scale, zooming can reveal a progression of semantic views, each with a physics of interaction particularly appropriate to that level. The key difference, and I think fundamental insight, is that computation enables design of physics not only to exploit our abilities and minimize our weaknesses, but also to allow us to directly interact with the semantic levels of tasks. Computationally based physics can not only mimic the physics of the world and thus exploit our knowledge of the world; they can also operate in ways that better match our abilities.

The early work of Furnas on generalized fish-eye views was especially influential for me and remains exceptionally relevant [15]. One deep insight was that instead of mimicking the physics of optics, a computationally based lens could compute a degree of interest function to determine what information is to be visible and at what scale. This, for me, is the canonical description and first example of semantic zooming. More important, and one of the reasons I am drawn to a physics characterization, is that not only can what we see be computed to be appropriate to various tasks but how we can interact can also be dynamically adjusted to task and context. Just as collaborative filtering (note that the PageRank algorithm is really a generalized fish-eye degree of interest function) coupled with massive computational power has radically improved the way we search, we need to explore how all important tasks can be improved and restructured by the design of cognitively convivial physics of interaction. Zooming and especially semantic zooming provides a glimpse of one dimension of this future design space.

ENDNOTES

[1] Ludwick Fleck gives a wonderful and amazing example of the history of an idea in *The Genesis and Development of a Scientific Fact.* University of Chicago Press, Chicago, 1979.

[2] Hollan, J.D., Hutchins, E., and Weitzman, L. Steamer: An interactive inspectable simulation-based training system. *AI Magazine* (1984), 15–27.

[3] Moboard was a computer-based training system for radar navigation that Ed Hutchins and I designed. It incorporated insights from Ed's earlier ethnographic studies, a graphical micro-world in which the student could explore the relationships between relative and absolute motion, and a tutorial facility that allowed the student to move step-by-step through radar navigation procedures. This system reduced the failure rate in radar navigation courses from 30 to about 3 percent at the Operation Specialist School in San Diego, CA. A reworked version of this program subsequently became standard refresher training for radar navigation aboard every ship in the U.S. Navy.

[4] Hollan, J., Rich, E., Hill, W., Wroblewski, D., Wilner, W., Wittenburg, K., and Grudin, J. An introduction to HITS: Human Interface Tool Suite. In *Intelligent User Interfaces.* S. Tyler and J. Sullivan, eds. ACM, New York, 1991, 293–337.

[5] Hutchins, E.L., Hollan, J.D., and Norman, D.A. Direct manipulation interfaces. *Human Computer Interaction 1*, 4 (1985), 311–338.

[6] Shneiderman, B. Direct manipulation: A step beyond programming languages. *Computer 16,* 8 (1983), 57–69.

[7] Card, S.K., Robertson, G.G., and Mackinlay, J.D. The information visualizer, an information workspace. *CHI '91 Proc. of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (New Orleans, LA, April 28-May 2). ACM, New York, 1991, 181–186.

[8] We changed the name to AR3T at the insistence of Bellcore lawyers and their concern about commercial systems named ART. We told folks that the "3" was silent and continued to pronounce it as ART.

[9] Witkin, A., Gleicher, M., and Welch, W. Interactive dynamics. *SIGGRAPH Comput. Graph. 24,* 2 (1990), 11–21.

[10] Perlin, K., and Fox, D. Pad: An alternative approach to the computer interface. *Proc. of the 20th annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA, Aug. 2-6). ACM, New York, 1993, 57–64.

[11] In the 3-D system we provided navigation via a mouse in the dominant hand and a six-degrees-of-freedom spaceball device in the non-dominant hand but still confronted classic 3-D navigation difficulties.

[12] Furnas, G.W. Generalized fisheye views. *CHI '86: Proc. of the SIGCHI conference on Human factors in computing systems.* (Boston, MA, April 13-17). ACM, New York, 1986, 16–23.

[13] Bederson, B.B., and Hollan, J.D. Pad++: A zooming graphical interface for exploring alternate interface physics. *Proc. of the 7th Annual ACM Symposium on User Interface Software and Technology. (*Marina del Rey, CA, Nov. 2-4). ACM, New York,1994,17– 26.

[14] Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing* 7 (1996), 3–31.

[15] Furnas, G.W. A Fisheye Follow-up: Further reflections on focus + context. *CHI '06 Proc. of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal, Canada, April 24-27). ACM, New York, 2006, 999–1008.

ABOUT THE AUTHOR Jim Hollan is professor of cognitive science at the University of California, San Diego. He co-directs the Distributed Cognition and Human-Computer Interaction Lab with Ed Hutchins and the ubiquitous computing and social dynamics research group with Bill Griswold and Barry Brown. His research explores the cognitive consequences of computationally-based media, with interests spanning across cognitive ethnography, distributed and embodied cognition, human-computer interaction, multiscale information visualization, multimodal interaction, and software tools for visualization and interaction. His current work involves three intertwined activities: developing theory and methods, designing and implementing prototypes, and evaluating the effectiveness of systems to understand the broader design space in which they are situated. He is a member of the ACM CHI Academy and in the past led the Intelligent Systems group at NPRDC and UCSD, directed the HCI Lab at MCC and the computer graphics and interactive media research group at Bellcore, and served as chair of computer science at the University of New Mexico.

DOI: 10.1145/1897239.1897255