

The Reification of Metaphor as a Design Tool

ALAN F. BLACKWELL
University of Cambridge

Despite causing many debates in human-computer interaction (HCI), the term “metaphor” remains a central element of design practice. This article investigates the history of ideas behind user-interface (UI) metaphor, not only technical developments, but also less familiar perspectives from education, philosophy, and the sociology of science. The historical analysis is complemented by a study of attitudes toward metaphor among HCI researchers 30 years later. Working from these two streams of evidence, we find new insights into the way that theories in HCI are related to interface design, and offer recommendations regarding approaches to future UI design research.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; D.2.2 [**Software Engineering**]: Design Tools and Techniques; I.3.6 [**Computer Graphics**]: Methodology and Techniques

General Terms: Design, Human Factors

Additional Key Words and Phrases: UI metaphor, history of HCI, design research

1. INTRODUCTION

The idea of metaphor in user interface (UI) design has a troubled history, and an uncertain status today. The term “desktop metaphor” is widely associated with the success of the graphical user interface (GUI), and with the world’s most successful software product, Microsoft Windows. Use of the term arose when Research and Development groups at Xerox, then Apple, were inventing the modern UI. Many popular design guides, tutorials, and textbooks have since described metaphor as a central principal of UI design. However, during the twenty years subsequent to the release of the Apple Macintosh in 1984, many expert commentators have described metaphor as over-rated and even harmful. Recent textbooks warn against reliance on metaphor, and criticize the designs that were heavily influenced by metaphor through the 1980’s and 1990’s.

This research was supported in part by the National Science Foundation, Microsoft Research, the Medical Research Council, the Engineering and Physical Sciences Research Council, the Hitachi Advanced Software Centre, the Cambridge-MIT Institute, and Boeing Corporation.

Author’s address: A. F. Blackwell, University of Cambridge Computer Laboratory, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD United Kingdom; email: Alan.Blackwell@cl.cam.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2006 ACM 1073-0616/06/1200-0490 \$5.00

This article is not intended solely as a history of UI metaphor,¹ but additionally as an analysis of the process by which an idea has become a design tool. This is a process of reification (making abstract ideas more concrete) in two different ways: Ideas can be turned into generic “tools” in the form of techniques or devices a designer should apply, or they can be turned by designers into specific products, such as the original Macintosh. Metaphor is a fascinating case study of such reification because it has been a close contemporary, and in many ways a central theme, of the expansion of human-computer interaction (HCI) as a design research discipline. The method used here to explore this process is best described as historical sociology, inspired by Mackenzie’s study of mathematical proof in computer science [2001]. The results will hopefully inform future HCI research, as we continue to turn theories into design tools, by giving an opportunity to reflect on our own professional practice.

The structure of the article is as follows. After some preliminaries (reviewing the main question, describing my research methods, and offering a working definition of metaphor), the article is divided into three main sections. The first is principally historical, and focuses on the intellectual origins of the desktop metaphor. Although the basic story is well-known, I describe several new insights. The second section analyzes various commentaries on the implications of this metaphor. Although these debates have often focused on the desktop itself, they were always understood to have implications for universal principles of design. I similarly refer to the desktop throughout this article, but it should be understood that the desktop metaphor serves as a metonym—a particularly salient example to represent the whole set of metaphorical interfaces—and my analysis is equally relevant to many contemporary design issues for the web, mobile devices, and ubiquitous computing.² The final section focuses on the future of interaction design, particularly the kind of reification that results in recommended tools, methods, or structures of design practice for the construction and critique of other interface metaphors.

1.1 The Status of Metaphor

The word “metaphor,” like the word “affordance,” is very familiar to HCI researchers and practitioners. The concept of affordance (Norman [1988], adapted from Gibson [1979]), meaning the possibility for action that we perceive in a situation,³ is the target of regular complaints from researchers about the generality of the concept, its theoretical accuracy, and its applicability to design work [McGrenere and Ho 2002; Bærentsen and Trettvik 2002; Torenvliet

¹There are already some fine histories of HCI, providing more historical detail. These include histories of specific projects and laboratories, as well as historical notes and personal memoirs describing the contributions of specific inventors and innovators. I cite these where possible, and apologize to those not mentioned by name. For an authoritative account regarding primacy of invention, I am indebted to Brad Myers’ review of major innovations in HCI technology.

²Where an argument in this article is applicable more specifically to the desktop metaphor itself, I indicate this by placing the word “desktop” in quotation marks.

³The word “affordance” is itself part of the problem—if English scientists still used scientific German, I might have been able to use a more descriptive term like *Interaktionsmöglichkeiten* (do-with-ability-ness).

2003]. Nevertheless, professional designers employ it as an unproblematic design property [Norman 1999]. “Metaphor” is at least an everyday word, so can be used even more readily than affordance, despite the problematic history that will become clear in this article. Popular HCI textbooks give straightforward advice, as follows: “Designers of systems should, where possible, use metaphors that the user will be familiar with.” [Faulkner 1998, p. 89]. “Metaphors are the tools we use to link highly technical, complex software with the user’s everyday world.” [Weinschenk et al. 1997, p. 60]. “Select a metaphor or analogy for the defined objects . . . real-world metaphors are most often the best choice.” [Galitz 1997, p. 84]. “Real world metaphors allow users to transfer knowledge about how things should look and work.” [Mandel 1997, p. 69]. “Metaphors make it easy to learn about unfamiliar objects.” [Hill 1995, p. 22]. “Metaphors help users think about the screen objects much as they would think about real world objects.” [Hackos and Redish 1998, p. 355]. “Very few will debate the value of a good metaphor for increasing the initial familiarity between user and computer application.” [Dix et al. 1998, p. 149].

Commercial software design guidelines have been equally direct. Chapter 1 of the Macintosh human interface guidelines stated “[y]ou can take advantage of people’s knowledge of the world around them by using metaphors to convey concepts and features of your application. Use metaphors involving concrete, familiar ideas and make the metaphors plain, so that users have a set of expectations to apply to computer environments.” [Apple Computer Inc. 1992]. The Windows interface guidelines advised that “[f]amiliar metaphors provide a direct and intuitive interface to user tasks. By allowing users to transfer their knowledge and experience, metaphors make it easier to predict and learn the behaviors of software-based representations.” [Microsoft Corp. 1995].

However, the unproblematic nature of these recommendations is belied by complaints from researchers, and by designs that have failed after undue reliance on metaphor. When the first metaphorical “desktops” and “windows” became profitable, corporate HCI researchers built even more overtly metaphorical UIs, such as General Magic’s *Magic Cap* in 1994 and Microsoft’s *Bob* in 1995. The justification for these expensive development efforts and product releases, predicated on the importance of metaphor, was so convincing that the developers were surprised when extra-realistic pictorial metaphors did not succeed to the same extent that the relatively abstract “desktop” and “windows” metaphors had. However, the failures had been anticipated by earlier innovators such as Ted Nelson:

“What I object to is severalfold: first, these mnemonic gimmicks are not very useful for presenting the ideas in the first place; second, their resemblance to any real objects in the world is so tenuous that it gets in the way more than it helps; and third . . . the metaphor becomes a dead weight . . . The visualizations become locked to some sort of continuing relation to the mnemonic. It becomes like a lie or a large government project: more and more things have to be added to it”

[Nelson in Laurel 1990, p. 236].

This indictment appeared in *The Art of Human-Computer Interface Design*, a 1990 book commissioned by Joy Mountford, a senior manager at Apple

Computer who in the same volume justified the company's reliance on metaphor by quoting Gasset: "The metaphor is perhaps one of man's most fruitful potentialities. Its efficacy verges on magic, and it seems a tool for creation which God forgot inside one of His creatures when He made him." That book, including both praise and condemnation of metaphor, marked the middle of a decade (1985 to 1995) in which researchers anticipated problems with metaphor at the start and had experienced failure by the end. In the wake of commercial disappointment from designing overtly metaphorical UIs, textbook authors are now more circumspect. More recent textbooks, if they mention metaphor at all, warn that it should be treated with care. Some campaign against it, for example: "Searching for that magic metaphor is one of the biggest mistakes you can make in user interface design" [Cooper 1995, p. 53].

In reviewing the popular attitude to metaphor, I do not want to suggest that researchers were once confused, or that a once-accepted theory has now been "debunked." Most of the authors cited previously undoubtedly had a nuanced understanding of metaphor, and its relevance to UI design. In particular, the theories of conceptual metaphor that I describe next, first popularized by Lakoff and Johnson in 1980, are now widely understood as providing a productive model of abstract thought as an embodied phenomenon. This theory is clearly relevant to the visual representation of computer abstractions in a GUI, as we shall see. Furthermore, increasing understanding of HCI as a design discipline has led to recognition of the way that metaphor can function as a creative tool for the designer. In addition to these views, we recognize that the term "metaphor" has itself become a dead metaphor, a jargon reference to a particular set of design practices. Working designers and users have little interest in the origin of jargon terms, what debates they incurred, and who was right or wrong. However, we might at the least be concerned about "theoretical hygiene"⁴—whatever working principles designers and users adopt, researchers ought to know how we arrived at these understandings, and hopefully anticipate where we are going next.

1.2 Studying Reification

This research started subsequent to my own PhD, which attempted experimentally to assess the cognitive benefits of using metaphor in diagrams. At that time I had read extensively, and probably understood the accepted position on metaphorical design as well as most HCI researchers. However, after some years, I became uncomfortable with the way in which understanding of metaphor has changed since its invention, sometimes without clear justification for those changes. This led to the current research, and my interest in the process of reification, because of the way that changes in the understanding of metaphor have resulted in reification, and in changing practices and products.

My source materials for this project include previous histories, published research documents, online resources, interviews, and field recording. Publications and online sources are cited in the usual way. Interviews have been

⁴A term used, perhaps with some self-disparagement, by philosopher Brian Cantwell Smith [1996].

conducted with several of those who were closely involved in the development of GUIs and theories of metaphor. In most cases, those concerned also published more or less contemporary accounts of their work. The purpose of interviews was therefore to check facts, and reflect on less formal aspects of the work not described in published accounts. Those interviewed are named in the Acknowledgements.

Finally, I have drawn on field recording and transcription of a one-day workshop on metaphor in HCI, which was convened in September 2003 at the British Computer Society HCI conference by Mark Treglown. The participants are listed in the Acknowledgements of this article. More details can be found in the workshop report [Treglown 2004]. Mark sadly died in 2004. We owe him a great deal of gratitude for his dedication to this topic.

1.3 A Working Definition of “Metaphor”

Defining metaphor is not simple. David Leary introduces a major book on the subject by quoting Soskice: “[A]nyone who has grappled with the problem of metaphor will appreciate the pragmatism of those who proceed to discuss it without giving any definition at all” [Leary 1990, p. 4]. Nonetheless, in common usage, metaphor is regarded simply as a literary term. It is an example of a trope, or figure of speech, a tactic in which the writer helps the reader to understand one thing by describing it as though it were another, for example: “my daughter is a peach.” Traditional analysis would call the target domain of description (my daughter) the *topic* of the metaphor, and the source domain (a peach) its *vehicle*.⁵ Although simple at this level, the difficulty of finding an adequate definition comes from the fact that philosophers and academics have applied the notion of metaphor in many other ways, to explain mental processes and other objects of interest.

When used in HCI, a computer UI might be considered to be a kind of “literary” description; a representation created to help the user understand the abstract operation and capabilities of the computer. These abstract capabilities are therefore presented as though they were something else that the user might already understand. A command “menu” can be understood by analogy to a customer choosing one of the dishes listed on the menu in a restaurant. A dialog “button” can be understood by analogy to pressing a button on a control panel. A “carriage return” helped typists understand how a word processor works because this is what a typewriter did (when you reached the edge of the page, the carriage mechanism moved back to its original position). A “scroll” describes a multipage document in which the bottom of each page (originally parchment or papyrus) is glued to the top of the next. Constructing effective metaphors is to some extent a “literary” skill because it depends on the talent and insight of designers to see new analogies, perhaps drawing on theoretical research findings in order to choose the right set of correspondences. Advice on how to do so is given in texts such as those by Horton [1994], who offers

⁵Rather than these traditional literary/philosophical terms, I prefer the convention more common in HCI, which is simply to use the terms “source” and “target” to refer to each domain. They are easier to remember, although philosophically inexact.

many practical examples (e.g., use of a paperclip icon to indicate the operation of attaching documents together).

The preceding account of metaphor might easily be found in an HCI reference book. It accords with what many computer professionals believe, and would be accepted as reasonably uncontroversial by many professional interaction designers and HCI researchers. And if that were the whole story, this article could end right now! However, when one thing is described in terms of another, a third thing has been created: the relation between the two. An ancient BBC radio gameshow asks contestants to sing one song to the tune of another. The words of the first and the tune of the second are innocuous, but the juxtaposition is hilarious (e.g., the words of “I Am Sixteen Going on Seventeen” from *The Sound of Music*, sung to the tune of “The House of the Rising Sun”).

When we consider the UI metaphor, it is clear that many people were already able to understand and use computers before the arrival of the GUI. Nor is there anything complex about the other side of the analogy: desktops, menus, buttons, and scrolls as we understand them in the physical world. But just as with songs, it is the juxtaposition of these two relatively straightforward things that is enlightening.⁶ In the rest of this article, I ask what it is that we are actually describing by such a juxtaposition. When we propose that a computer be presented as a metaphorical office or typewriter, one of the things we are really describing is the intended user of this computer, describing him or her as an office worker or typist. When we designer-researchers in HCI imagine a UI to be a work of literature, we are describing ourselves as creative authors rather than mundane technologists. The relationship between users and designers structures the commercial and social context of HCI, and is the basis of our academic and professional discipline. The remainder of this article is structured to address these concerns, first in the historical context of desktop development, then commentary on the desktop and other metaphors, and finally, proposals for the way that metaphor should be used as a tool in future.

2. INTELLECTUAL ORIGINS OF THE DESKTOP METAPHOR

This part of the article is primarily historical, and presents two different perspectives. The first two subsections offer a brief review of the context in which the UI metaphor was originally developed: a description of the organizations where UI researchers were working, followed by a review of the philosophy of metaphor current at that time. Much of this has been described elsewhere, so readers already familiar with the basic facts may prefer to skip the next two subsections. The remaining subsections in this part subsequently investigate different conceptions of the user that informed these developments:

⁶The philosophical writing of C.S. Peirce deals at great length with the question of third things. An interest in such juxtapositions is also apparent in the fertile poststructuralist meditations of Deleuze and Guattari [1980], whose discussion of “rhizomes” draws attention to the points in a network of conceptual relationships where the relations themselves might be viewed as nodes. Many attempts to explain the relation between words and tunes can be found at <http://www.isihac.co.uk/games/ostttoae/>.

conceptions arising from anthropology, cognitive psychology, computer science, and education.

2.1 Organizational Context of Technical Research

The circumstances surrounding the original development of the desktop metaphor have attracted considerable historical attention, prompted by its subsequent commercial and cultural impact. Book-length popular histories include Smith and Alexander [1988], Levy [1994], and Rheingold [2000]. More specialized treatments include Myers' [1998] review of interface technologies, Kay's [2002] and Müller-Prove's [2002] reflections on the GUI, and accounts of particular laboratories and projects, such as Johnson et al. [1989], Raskin [1994], Kay [1996], Liddle [1996], Perkins et al. [1997], and Tesler [2004].

It is clear that early HCI research in the 1960's and 70's built on a legacy of leadership from influential research policymakers such as Vannevar Bush [1945] and J.C.R. Licklider [1960], who envisioned a future of human augmentation, even "symbiosis," with computers. Cold War investment in technological research could fund long-term research efforts toward these ends. The large program at the Lincoln Laboratory [Buxton 2005] resulted in the first GUI, Sketchpad [Sutherland 1963] and a primordial 3-D mouse, the Lincoln Wand. Research at the RAND Project [Ware 1996] resulted in the JOSS and GRAIL end-user programming systems and the RAND graphics tablet. These integrated hardware and software initiatives funded by defense grants were often described using military scenarios, so that Bush's vision of a scholar navigating the world's knowledge via the proto-hypertext Memex [Bush 1945] evolved into an intelligence analyst or General reviewing, consulting, and commanding via multimodal computer interfaces. Examples include Engelbart's oNLine System (NLS) project at SRI that was described as augmenting the human capabilities of research managers in technocratic scenarios [Licklider 1968], while the MIT "Dataland" spatial data management system (SDMS) offered an aerial command perspective of naval deployment [Herot et al. 1980].

These research efforts were funded within a military tradition of creating integrated project teams, bringing together whatever research skills and resources were required for technical progress. The requisite skills in this case included a concern with human factors, extending to interest in the cognitive demands placed on the operator. This tradition of well-funded strategic research teams was continued in the corporate research laboratories of IBM and Xerox. Xerox employed a diverse range of talented researchers who we now recognize for their contributions as creative visionaries (Alan Kay and David C. Smith), cognitive psychologists (Stuart Card and Tom Moran), interaction designers (Larry Tesler and William Newman), and of course, talented systems engineers and application developers (including Butler Lampson, Chuck Thacker, and Charles Simonyi). Various members of these teams moved on to Apple as developers of the Lisa and Macintosh (where the groundwork had been prepared by Jef Raskin), and later to Microsoft. There was a well-documented flow of intellectual influence between key projects, from Sketchpad and NLS to the Xerox Alto, from Alto Smalltalk and SDMS to the Xerox Star and Apple

Lisa, with contemporary projects at RAND and Lincoln Lab well-known and influential across the research community.

There have been occasional differences of opinion about who “invented” aspects of the modern desktop, but it seems clear that the main elements arose by the research community drawing together and accumulating successful innovations. Several groups used pictograms, some calling them “iconic” and some not.⁷ The screen was partitioned into graphic regions based on the “clipping windows” used in GRAIL and NLS, and this principle was extended in Smalltalk (as well as providing the technical origin of the trademark “Windows”). It is unclear who first noticed the resemblance of the Smalltalk display to overlapping pages on a desktop [Levy 1994; Johnson et al. 1989]. The insight that office work might be facilitated by depicting objects from an actual office has been the subject of legal dispute,⁸ and may have been discovered independently at Xerox [Smith et al. 1982], Apple [Perkins et al. 1997], and IBM [Schild et al. 1980].

Whatever the genesis of individual features of the modern GUI, these were not inevitable discoveries of natural usability principles. On the contrary, most of these innovations were driven by designers appropriating hardware and software techniques, applying them to the UI on the basis of intuition, with user testing applied as much to arbitrate in design disputes as to discover requirements [Morgan et al. 1983]. A central and profound invention was Kay’s conceptualization of the GUI as a new kind of end-user programming language. Many features of Alto and Star were influenced by the Smalltalk concepts of object-oriented programming. Icons (especially in Smith’s Pygmalion [1977]) were direct representations of Smalltalk objects, user actions on these objects were conceptualized as method invocations, generic command menus as polymorphic operations, and object state could be inspected via property sheets (these later evolved into dialogs and application windows).

The most dramatic user benefits arose from the development of direct manipulation, a term that we now attribute to Shneiderman [1983], although it was already used informally with the same meaning (e.g., the Dataland team describe the benefits of “directly manipulating” an object on the screen to control its state with immediate feedback [Donelson 1978]). Direct manipulation principles were not discovered at one step, but developed through a continual series of innovations, including selection of an object by pointing (in Sketchpad), separating text insertion from replacement by using a cursor between characters (pioneered in the TECO real-time edit mode, then applied in Tesler

⁷To a reader of C.S. Peirce, any picture of an object appearing on a display would be considered an *icon* referring to this object. Tesler says he was aware of this usage (from a semiotics book, although he cannot recall which) as early as 1973, when it was used in a Xerox memo by Rulifson and Tesler. Smith [1977] said he was inspired to coin the term by analogy to religious icons. However, many of the “icons” now used in GUIs would more properly be called *indices* or *symbols*, if using semiotic terminology.

⁸Apple Computer, Inc. versus Microsoft Corp., 35 F.3d 1435 [1994] was a copyright infringement lawsuit in which Apple Computer sought to prevent Microsoft and Hewlett-Packard from using visual GUI elements that were similar to those in Apple’s Lisa and Macintosh operating systems. From Wikipedia, http://en.wikipedia.org/wiki/Apple_v._Microsoft.

and Mott's Gypsy, based on Peter Deutsch's insight of how to select an insertion point with the mouse), using pictorial icons as representatives of abstract relations (Pygmalion and SDMS), and relocating objects by "drag and drop," rather than invoking a "move" command (in the Lisa re-engineering of their original "Filer" into the "Desktop Manager"). Despite the later significance of these developments, they were not necessarily experienced as being intuitive at the time. Independent evaluation of the first Lisa in 1983 observed that users found most aspects of the desktop metaphor to be extremely confusing, with windows, icons, and direct manipulation all proving frustrating to a sample of intended target users [Carroll and Mazur 1986]. If there is any underlying cognitive advantage from these basic operations of pointing, selecting, and moving icons, we might now attribute it to the "specificity" of visual representations having less expressive alternative interpretations [Stenning and Oberlander 1995] or to the opportunity to envision action consequences by avoiding the "frame problem" (determining the scope of resulting change [Lindsay 1988]). However, these theoretical perspectives were not available at the time.

2.2 Theoretical Context of Ideas About Metaphor

As apparent from this brief review, the desktop metaphor was not directly motivated by any coherent theoretical stance regarding metaphor, but was chiefly a product of technical evolution and historical contingency [Gentner and Grudin 1996]. Liddle [1996, p. 21] states (contrary to later design advice) that there was no intent in Star to imitate physical objects⁹, but simply to present pictures that users could relate to their jobs. However, a critical vocabulary was necessary to interpret these innovations, and theories of metaphor rapidly developed. John Carroll, in several joint publications including a chapter in a popular HCI handbook [Carroll et al. 1988], classified three general approaches to metaphor in HCI: The *operational* approach describes metaphor as an educational resource for the user. The *structural* approach evaluates specific relationships between source and target domains of the comparison. The *pragmatic* approach anticipates breakdowns and mismatches between different user goals and the presentational metaphor, regarding the mismatches as an opportunity for richer experience. Subsequent reviews of metaphor in HCI [e.g., Wozny 1989; Madsen 1994; Treglown 1999] continued to be influenced by Carroll's classification, but an equally significant influence came from a philosophical development in the understanding of metaphor around the same time.

The classical philosophical understanding, at least since Aristotle's *Poetics*, was that metaphor describes one thing as though it were something else. Black [1962, 1993] challenged this *comparison* model of metaphor, proposing an alternative *interaction* model in which the reader constructs new meaning out of the differences between literal and figurative interpretations, rather than their similarities. This creative process of interpretation provides opportunities for new insight, in circumstances such as scientific discovery [Hesse 1966]. Indeed, my own experiments found that software metaphors could improve learning

⁹As occurs in "skeuomorphism": the preservation in new products of nonfunctional features from previous generations [Coyne 1995, p. 283].

when users construct their own meanings from dissimilar elements, rather than being shown systematic and consistent mappings of the kind recommended in HCI textbooks [Blackwell 2001]. Recent work by Hsu [2006] confirmed my result, finding that novices with poor prior understanding of a domain may perform even worse when given an explanatory metaphor. A philosophical stance in support of these experimental findings is that of Davidson [1984], who questions whether a metaphor can have any objective meaning at all in the interaction model, proposing that the only reasonable basis for describing a metaphor is in terms of the effect that it has on the listener. The philosophies of Hesse and Davidson, and their application to concepts of design, are discussed at greater length in Coyne [1995].

Both the comparison and interaction models were challenged, around the time that the desktop GUI was developed, by a new theory of metaphor. This *image-schematic* model of metaphor has chiefly been the work of George Lakoff and Mark Johnson [Lakoff and Johnson 1980, 1999; Johnson 1987; Lakoff 1993]. Their claim is that all meaning is grounded in bodily experience, that any abstract concept can be shown to depend on mental images, and that this is revealed by the pervasiveness of spatial and physical metaphors in our vocabulary. Hundreds of published examples include schemas such as COMPLIANCE IS FOLLOWING, as in “you ought to go along with this argument,” or A PROBLEM IS A TANGLE, as in “you may need to unravel this argument.” Other philosophers had already noted that ordinary language is pervaded by dead metaphors (e.g., Nietzsche in 1873, “illusions of which we have forgotten that they are illusions”). Philosophical abstractions are themselves metaphorical (e.g., the word “abstract” itself is from the Latin expression “to pull away from”).

Despite the familiarity of metaphorical language, the strong claim by Lakoff and Johnson that all abstract meaning is derived from images was striking to HCI researchers, who were themselves at the time designing pictorial images to express computer abstractions. Although the jargon of computer science (e.g., “open,” “save,” “paste”) already depended completely on metaphorical language to express new abstract concepts [van Dyke 1988; C. J. Johnson 1994], Lakoff and Johnson’s work seems to have prompted awareness of metaphor as beneficial to nontechnical users. Prior to this, usage of the word “metaphor” in HCI had been haphazard. Howe and O’Shea [1979] described the computer as a metaphor for mathematics, rather than vice versa. Goldberg and Rosson [1979] introduced as a metaphor for designers the “filtering template,” a proposed user interface component that we would now describe as a software design pattern. Meanwhile, Smith’s work on Pygmalion [1977] and the Xerox Star [Smith et al. 1982] had used the word metaphor in various ways, including the computer as a metaphor for the office (not vice versa), and visual images as a metaphor for human thought.

Lakoff and Johnson’s [1980] work had reached the HCI community by the time of the Gaithersburg conference that inaugurated the *ACM CHI* series. Halasz and Moran [1982] were concerned about instructional analogies, such as comparing disk storage to a filing cabinet (as done in the Xerox Star). They felt that the differences between computers and cabinets would confuse users. Their Gaithersburg article *Analogy Considered Harmful* argued from Lakoff

and Johnson's work that UIs should employ less precise "literary metaphor" so as to prompt user understanding without inviting detailed comparison. This strategy is a move from Carroll's structural to operational classes of metaphor. However, it is not a particularly accurate representation of Lakoff and Johnson, who claim that all knowledge, not just "literary" language, is based on image-schematic metaphors. In fact, later writers in HCI, such as Erickson [1990], argued that metaphor is important for design precisely because Lakoff and Johnson show it to be pervasive in thought. However, if it is true that metaphor underlies all human thought, do designers have any choice about whether users will interpret their designs metaphorically? I return to this question later, when discussing possible methodologies for metaphorical design.

As described in the next section, educational/operational models had significant influence on the early development of the desktop metaphor. However, critical discussion since then has offered a wide range of purely structural critiques. A common theme has been to suggest that other literary tropes might provide more appropriate structures, perhaps applied alongside metaphor (e.g., [Marcus 1994; Noble et al. 2001; de Souza et al. 2001; de Souza and Barbosa 2006]), or that design tools ought to enumerate and evaluate the specific elements of structural correspondence (e.g., [Anderson et al. 1994]). These structural approaches are hard to justify in terms of Lakoff and Johnson's theory. Their claim that all meaning relies on embodied experience challenges the presumed correspondence between symbolic structures of computer science and human thought. In fact, Johnson [1987] questions the distinction between figurative and literal language on the grounds that an objective distinction itself requires a literal "god's-eye view" of disembodied rationality. Arguments around this topic tend to be awkward because regardless of whether we can justify an objectivist stance, we have no trope-free metalanguage in which to ground the comparison [Rotman 1999]. These are still central problems in HCI. At the workshop I attended in September 2003, the influence of Lakoff and Johnson on HCI metaphor was the single greatest issue of common concern, for example, from participants proposing structural accounts based on more recent work in cognitive semantics by Fauconnier [1997].

Metaphor is not a straightforward topic in philosophy. The second edition of Ortony's collection *Metaphor in Thought* [1993] is an invaluable reference for readers who wish to pursue these various perspectives further, but the remainder of this article focuses on the way that these theories have been adapted to HCI, rather than trying to establish a definitive position on metaphor in general. Carroll's structural approaches must be based on a coherent philosophical position, and critiques can be offered simply by adopting an alternative. More interesting debates have occurred around operational and pragmatic approaches, and these, roughly speaking, are the topic of the next few sections.

2.3 Choosing the User to Describe

Commercial success of the desktop interface rested to a large degree on the insistence that it should be designed for, and evaluated with, real users. Tesler found evidence to support his intuitive commitment to modeless interaction

after observing a secretary at Xerox PARC struggling with a command editor [Tesler 2004], and he subsequently organized a substantial program of user testing during the Lisa development at Apple [Morgan et al. 1983]. At IBM, Carroll spent hours observing users who were being “terrorized by word processors” [Carroll 1997]. However, this common concern with studying users was not matched by a common understanding of who the user population for these experimental computers might be. At a time when the word processor market was expanding rapidly, typists represented an important commercial opportunity, while also conveniently available for use as experimental subjects at any research laboratory wishing to compare novel computer interfaces to typewriters. Yet the Xerox Star and IBM Pictureworld projects were aimed not at lowly typists, but professional bureaucrats; “office professionals” [Schild et al. 1980; Smith and Irby 1998]. The Xerox Alto was conceived originally as a computer for children (Kay’s “interim Dynabook” [Kay 1972]) and then developed into a personal computer for computer scientists [Wadlow 1981], intending to compete with NLS [Lampson 1972]. Xerox also announced its intention to compete in the word processor market established by smaller companies such as Wang,¹⁰ so the System Development Division that hosted the Star project must have suffered considerable dissent in pursuing these diverse markets.

2.4 Describing the User as a Cognitive System Component

From an engineer’s perspective, understanding the intended user is a crucial prerequisite for the definition and construction of a “user interface” (i.e., the aspect of the system that “interfaces” to the human user, in engineering terms). This is expressed most strongly in the tradition of cognitive systems engineering, which relies on “the model of a human information process viewed as a system component” [Rasmussen 1986]. At Xerox, cognitive psychologists Card and Moran, with external consultant Newell, also constructed a model of human performance that could be used by engineers [Card and Moran 1986]. Many aspects of the “model human processor” provided generalized descriptions of human performance that were reasonably uniform in various (able-bodied) alternative user populations (e.g., movement, perception, and short-term memory), but it was also necessary to supplement these behavioral parameters with a model of the users’ internal representations that would explain how they learn to use an unfamiliar interface. For both cognitive psychologists and engineers, this perspective framed the theoretical work that led to the desktop metaphor.

In the 1960’s and 70’s, cognitive psychology was an exciting research field that was closely linked to computer science (especially artificial intelligence research) through a common concern with representation. Computers were a more convincing metaphor for mental processes than earlier philosophies of mind based on metaphorical clockwork or hydraulics. But the computer metaphor went both ways [Hoffman et al. 1990], with engineers using

¹⁰The Xerox 800 Electronic Typing system—Jonathan Grudin, personal communication, July 13, 2005.

descriptions of human mental processes to inspire new software designs.¹¹ The designers of early graphical user interfaces offered a variety of justifications for their work that were borrowed from experimental results in cognitive psychology. The Pictureworld authors justified the use of a visual interface by a “time-sharing” analogy of concurrent human visual and verbal processing based on the work of Paivio and Kintsch [Schild et al. 1980]. The Dataland designers were impressed by Neisser’s theory of spatial perception, and designed multimedia systems that were “eminently compatible with the human perceptual system” [Donelson 1978]. Until the advent of the *CHI* series, there was little opportunity to compare these theories of mind through publications at technical computer science venues. Nonetheless, it seems that the most general influences originated from cognitive perspectives on education via Piaget and Bruner ([1966], described in the next section) and from the study of *mental models*.

The description of the mental model as a user’s internal data representation was derived from several research streams. In Edinburgh, Young had developed a cognitive account of human learning by analogy to the construction of a symbolic model in machine learning [1976]. This comparison was then reversed in a study of humans who were themselves trying to understand machines: users of pocket calculators [1981]. The artificial intelligence perspective on human learning was an obvious candidate for HCI, but alternative mental model theories were also derived from research in cognitive anthropology, which was attempting to develop cultural grammars in the manner of Chomsky [Keesing 1987]. The kinds of human knowledge studied in such research were often technical and scientific, as in Kempton’s study of home heating controls [1987], or Gentner and Gentner’s interest in popular understanding of electric current [Gentner and Gentner 1983]. Research in both psychological and anthropological traditions allowed an account of the user’s understanding to be encapsulated by engineers in a black box, called the “mental model,” which could then be analyzed as an objective, formalizable, and symbolic entity [Collins and Gentner 1987; Collins 1994]. According to Lakoff and Johnson [1999], this *computational theory of mind* is itself a metaphor, evolved to help people like engineers talk about minds using technical terms that they already understand.

2.5 Designing Mental Models for End-Users

The key development leading to the desktop metaphor was not simply the explanation of user knowledge in terms of a computational model, but the suggestion that system designers could extend their programming work to construct such a model in the user’s mind. The design of a “user model” (or “conceptual model” in later writing) was a key element of the Xerox design strategy [Bewley et al. 1983]. It was developed initially as a component of Card and Moran’s work, and made explicit in a methodology for UI design published first in an internal

¹¹Some artificial intelligence researchers even suggested that they were creating literal minds in a computer, or that humans really were symbol processing machines. Reflection on the role of metaphor in science provides cause for doubt when, among multiple alternative metaphorical descriptions, the supporters of one alternative argue not only that it is a better description, but that it is *not a metaphor at all*.

Xerox document [Irby et al. 1977], then quickly popularized in a chapter of one of the first textbooks on user interaction design [Newman and Sproull 1979].

These computational descriptions of the user's mind were not intended to be reductionist or disempowering. On the contrary, users were described as beneficiaries of the user model, in a rhetoric of empowerment whereby the privileges of programmers and scientists would be extended to others. The contemporary publications resemble today's interest in "end-user" populations. Star was aimed at "casual users" [Bewley et al. 1983]. Kay's ambition in his PhD project FLEX was to create an end-user programming language akin to JOSS and LOGO, but fitted to a broader user base by using a semantics based on mental, rather than computational, models [Kay 1969, 1996]. It is the conceptual model of Kay's subsequent Smalltalk programming language, at least as much as an analogy to a physical office, that underlaid the Star and Lisa "desktops." Mayer hoped to assist people who would program computers at home [Mayer 1979], and Dataland aimed to provide computational abstractions "accessible to naïve users and even children" [Donelson 1978]. Both the rhetoric and products were greeted enthusiastically in hobbyist magazines such as Byte [Williams 1983], but the expected beneficiaries were slower to applaud, as when Carroll and Mazur [1986] set out to test the benefits for office professionals using the Lisa, but found that most experienced confusion and frustration, rather than empowerment.¹² This focus on end-users might have been laudable as the moral action of a privileged technocratic elite, but could also be criticized as promoting a deficit model of system users. The creation of a special end-user version of facilities enjoyed by computer scientists resembles the "public understanding of science" model whereby the public is presumed to have inadequate understanding, and to be unqualified to participate in informed debate unless further educated. Attempts to create natural and intuitive UIs based on nontechnical user models are as questionable from this perspective as the research programs that describe public "naïve models" of physics by contrast to professional (academic) experts [Wynne 1994].

If we contrast philosophical theories of metaphor to the intent of "user model" design, several problems are evident. One is that Johnson's account of metaphorical understanding as bodily, subjective, and imag(e)inative cannot possibly be contained in a computational representation [Johnson 1987]. Furthermore, although the mappings of a familiar metaphor might be regarded as predictable, "dead," or "lexically entrenched," the use of metaphor retains creative potential [Fauconnier 1997] that may lead to users' wild and unruly interpretations escaping the control of the designer's intent. This concern arose repeatedly at the 2003 workshop as an ongoing problem for metaphor designers. Several presenters described their continued application of research results from cognitive psychology in order to predict and improve the "efficiency"

¹²Tesler (personal communication, August 3, 2005) says that many of the difficulties observed by Carroll and Mazur could be attributed to the tutorial application LisaGuide, which had not been adequately tested. Tesler's own studies had shown that Lisa was more acceptable to "total computer novices," whereas Carroll's investigation placed Lisa at a disadvantage by observing users who had prior experience with conventional computers.

of the system interface to users. However, users are found to be unpredictable in practice. Rather than being compliant objects of analysis and the recipients of a designed mental model, metaphor users do not apply the set theories and database relations of computational knowledge structures. Instead, they treat metaphorical meaning as fuzzy, mutable, dynamic, and social. It is precisely these attributes of metaphor that appeal to and empower designers, as described to follow. Unfortunately, when practiced by users, these ways of understanding metaphor threaten and invalidate the design of the mental model as a system component, leading to the post-modern spectre of the “death of the designer” [Richardson 1993].

2.6 Cognitive Theories of Education: The User as Learner

The discussion of how we construct knowledge for others, with its attendant commentaries of power relations, creativity, and control, closely resembles debates in educational theory. This is no accident. Cognitive theories of education were a fundamental influence on the HCI research of the 1970's. In the first half of the 20th century, behavioral psychology might have been used to justify rote learning in the kind of educational context where drill-and-copy exercise was prevalent, but this was not an attractive image—the school as a place of educational enforcement. In the second half of the century, cognitive psychology, especially from the developmental theories of Piaget, described children in a more acceptable light. In the USA, philosopher Jerome Bruner derived a cognitive epistemology from Piaget's work, and wrote campaigning books, such as his *Theory of Instruction* [1966], recommending cognitive educational practices that focus on helping the student build internal mental representations, rather than rote behaviors. Where Piaget described developmental stages of cognition leading from concrete action-based competence to abstract linguistic competence, Bruner recommended this progression as a pedagogic strategy for all education, not just young children: Students should encounter new knowledge first through “enactive” physical activity, as a result of which they would form “iconic” sense impressions, and organize these into “symbolic” mental structures.

This creative process of “active learning,” popularized at the time as “discovery learning,” was attractive to schools, parents, and policy makers. It also appealed to innovators applying computers to education. Seymour Papert at MIT had worked with Piaget, and was influenced by Bruner's active learning theory in developing the LOGO language [Papert 1980]. Alan Kay cited Bruner's cognitive epistemology as motivating the FLEX and Smalltalk languages, and in recent lectures he has described Piaget and Bruner as providing the main philosophical motivation for the iconic GUI that developed from Smalltalk [Kay 1996, 2002] (however, Kay subtly reorders the enactive-iconic-symbolic progression of Bruner in a way that better fits the rhetoric of the GUI by substituting a screen icon for Bruner's sensory mental image in the formulation “doing with images makes symbols,” which rather implies iconic-enactive-symbolic progression).

We may pose philosophical questions about the validity of the theoretical progression from Piaget's developmental theory to Bruner's epistemology, then

Bruner's theory of education to Papert's adoption of active learning and Kay's development of the GUI programming language. But an even more interesting question is raised by the fact that Bruner was not the only popular cognitive theorist of education in the 1960's. David Ausubel was a vigorous critic of Bruner [Ausubel 1968]. Ausubel felt that the vogue for discovery learning was a fashionable error derived from the "progressive education" movement, popular because it supported a democratic illusion that all children could be creative thinkers. Ausubel calls this a questionable, even incredible, notion, saying that true creativity is a cognitive exception, and that the mental process of a Mozart has nothing to do with the education of an ordinary child. His educational theory, like Bruner's, emphasized cognitive structures. But Ausubel said these cannot be discovered by the child, and must instead be presented explicitly by the teacher through the expository presentation of "advance organizers" to which later facts can be connected. However, Ausubel was worried that his expository teaching would be viewed in the 1960's as being inappropriately authoritarian, associated with indoctrination of officially sanctioned truth. As noted by LeFrancois [1975], "discovery learning," whatever the virtues of the two theories, is a term that sounds better when used at a meeting of the Parent-Teacher Association.

Surprisingly, since Ausubel's notion of the advance organizer is based on principled opposition to Bruner's active learning, both ideas seem to have been absorbed into the cognitive theories surrounding the desktop metaphor. Advance organizers are mainly known in HCI through the work of Richard Mayer, who made many experimental studies following Ausubel's own experiments on teaching technical subjects, such as steel processing [1960]. Mayer initially addressed the teaching of statistics [Mayer and Greeno 1972], but soon turned to programming languages, including studies of FORTRAN [1975] and BASIC [1979]. After several publications in educational psychology venues, Mayer's findings were brought to the attention of the computer science community through publication in the *Communications of the ACM* [1979], after which the term "advance organizer" became common as an empirically justified formalization of the "user model" or "mental model" described earlier. At the Gaithersburg conference, Foss et al. [1982] described using an advance organizer to instruct novice users on a word processor, and Carroll recalls their research as an origin of his "operational" category of metaphor use. It is likely that the projects exemplifying Carroll's "pragmatic" category (which is in fact called "active learning" by Carroll et al. [1988]) were motivated quite differently by the discovery learning perspective, and it seems that those researchers were unaware of Ausubel's work,¹³ just as the mental model designers were apparently unaware of Bruner's.

The two incompatible cognitive theories of education that have been inherited by HCI may well have been played out as a long-term incompatibility in theories of the UI. The expository tradition of the advance organizer is clearly related to the suggestion that we "design the user model," as promoted by the cognitive psychology team at Xerox. When interpreted in the context of

¹³John Carroll, personal communication, May 6, 2005.

a computational theory of mind, and notions of the user as a cognitive system component, Ausubel's anxiety that expository teaching would be seen as authoritarian can indeed be extended to a rather nightmarish vision of UI designers engaged in mental engineering. In contrast, Kay's ambition to build a computer for creative use by children shares all the popular appeal of discovery learning, and is a clear antecedent for the eventual marketing of Apple Macintosh as the computer of choice for the creative industries.

At the time of writing, no cognitive theory of education is universally adopted in educational research. Although Piaget is honored as a founder of developmental psychology, his cognitive theories are studied mainly by Piagetian specialists. Similarly, Bruner and Ausubel are seldom cited in HCI. Even if they were better known, it is not certain that either developmental or educational theories are good descriptions of an adult using a GUI. Various researchers continue to design new programming languages for children, generally citing Papert's concepts of "constructivism" and "bricolage" (see, e.g., Turkle and Papert [1992]) as their motivating educational theory. Nevertheless, Ausubel's warnings regarding the unwarranted assumptions about creative ability in discovery learning remain relevant to HCI, especially considering the kind of individuals who applied Bruner's ideas. Alan Kay was an exceptionally talented child, frustrated by conventional schooling [Childress 1989; Lazere and Shasha 1995]. His KiddiKomp and DynaBook proposals [Kay and Goldberg 1977] are quite transparent in their adult enthusiasm to build a dream toy for the child genius. Articlet, who says he and Kay were the only researchers building computers for children in the 1970's, was also motivated by his own experience as an exceptional child. In *Mindstorms* [1980], he describes his childhood fascination with mechanisms, and intuitive understanding of geartrains, forming the inspiration for a mechanical understanding of computation through the LOGO turtle. Is Ausubel right that we cannot generalize from genius to ordinary cognition? If so, perhaps those who were exceptional children should not be trusted to design school computers. Furthermore, perhaps Apple's desktop metaphor, despite its success as a marketing icon of the creative counter-culture, should not automatically be credited with educational advantages, for either children or novice adult users. Indeed, some educational researchers have broader concerns about the way in which constructivism has advanced from a theory of knowledge to a general theory of pedagogy.¹⁴

3. REIFIED METAPHORS IN THE GROWTH OF THE DESKTOP

Moving forward from historical origins of the desktop metaphor, I now focus on the period after interactive graphical systems such as NLS, Dataland, Alto, and Lisa had become well-known. In particular, there was a surge of commentary on metaphor after the introduction of the Macintosh in 1984. Interest in the desktop and its implications has continued to the present day, and I

¹⁴Debbie Reese, personal communication, later published as Reese, D. D. (to appear). Moon-Pie in the sky? Engineering instructional metaphors within virtual environments. In J. K. Gilbert, M. Nakhleh and M. Reiner (Eds), *Visualization: Theory and Practice in Science Education*. Kluwer Academic Publishers, Boston, MA.

draw especially on material from my transcription of the 2003 *BCS* workshop. The 20 participants in this workshop were mostly academic HCI researchers, with one or two practitioners. Most had published research articles or books on the topic of metaphor, and all submitted position papers in order to attend. They ranged in seniority from PhD students to postdoctoral researchers and professors.

My source data from that workshop is a direct transcription of free, sometimes speculative, discussion. Not all statements were well-considered research judgments, hence unlikely to appear in a research publication. For this reason, I have not directly attributed statements to individuals, but present them as views from the workshop as a whole (of course, this is not intended to imply any consensus). The value of analyzing such informal statements alongside more conventional historical and academic evidence is that they are representative of routine conversation about metaphor among a group who can be considered reasonably familiar with current HCI research and practice, but who were not directly involved in the original development of the aforementioned ideas. The status of this data is therefore to provide evidence of the reification processes that surround UI metaphor. The following subsections are organized thematically, addressing the different varieties of reification identified from analysis of historical and contemporary commentary.

3.1 Reifying Metaphor as a Visual Communication Channel

Bruner, Ausubel, Papert, and Kay were all concerned with the way that learners acquire information. To a computer scientist, the acquisition of information implies the existence of a communication channel, and the technical nature of this channel was discussed by Kay, by his student David Canfield Smith [1977], and by the Dataland and Pictureworld teams. In particular, pictorial or diagrammatic screen representations were viewed as efficient carriers of instructional information. In the 1970's, there was strong interest in visual modes of thought, although detailed cognitive models of diagrammatic reasoning did not become widespread until the 1990's (e.g., [Glasgow et al. 1995; Blackwell 2001]). Kay and Smith cited models of creative visualization from the arts (e.g., Koestler and Gombrich), while Bill Verplank joined the Star project after working on an experimental Stanford course in visual thinking [McKim 1980]. Concern with visual literacy and media culture was associated with interest in the alternate cognitive styles of visualizers and verbalizers, with right brain/left brain asymmetry, and with the distinction between symbolic and analogical modes of thought. These intuitions regarding visual information motivated not only UI design, but software engineering innovations such as Chen's entity-relationship diagrams [Chen 2002], and the long-standing pursuit of visual programming languages [Blackwell 1996].

At the 2003 workshop, metaphor was described as a visual communication channel via which the designer achieves the rapid transfer of an effective mental model into the user's head. This channel model implies that the novice user might be a comparatively passive recipient of new understanding, and the metaphor a static carrier of information. Active, subjective, contextualized, or

embodied interpretations of visual metaphor are inconvenient because users might discover or construct new interpretations, rather than receiving the expected model [Richardson 1993]. Furthermore, we might expect that new users would gain the most information, and that experts would gain relatively little benefit. The design ideal in this case is to provide effective *access* to this information, while allowing expert users to *bypass* the metaphor if they already have the necessary information. This was indeed the expectation of many commentators during the 1980's, when advocates of PC/DOS argued that the Macintosh was not suitable for expert users.

Where metaphor is employed in educational software, as studied by one workshop participant, this perspective can lead to treatment of the visual presentation channel as if it were a kind of sugarcoating disguising relatively unwelcome (to children) educational content. From the teacher's point of view, this visual "front-end" is a barrier separating students from the text they are expected to absorb. Meanwhile, children subvert the designer's intention by "playing" the navigational interface as if it were a videogame, in effect sucking off the sugarcoating and then spitting out the content pill. These tensions are by no means exclusive to educational software. Products such as Microsoft Bob did attempt to provide special novice visualizations that could be bypassed by expert users, with the result that they were perceived as frivolous or condescending. These experiences do not necessarily invalidate the concept of educational metaphor in the UI, but they certainly cast doubt on the common engineering perception that an instructional metaphor might be retro-fitted as a communication channel that explains existing system functionality, rather than being the core of the design.

3.2 Reified Descriptions of Self-Augmentation as Heroic Technology

With the exception of work by Papert and Kay, expensive computer prototypes in the 1960's and 70's were not used by children. Neither were they used by office staff, other than in experimental situations. Instead, the users of the systems were the researchers themselves, and the military/industrial research funders to whom they demonstrated their products. This class of user is far less likely to be described as naïve, novice, or passive recipients of a "mental model." Computer science researchers spend their time reading and writing research articles, programming, and discussing their work with other researchers. The paradigmatic system concept to support this intellectual work was Bush's proposal for a Memex: a complete research library miniaturized into a desk. Licklider envisioned the researcher and his knowledge tools forming a man-computer symbiosis, and Engelbart's NLS made significant progress toward this vision [Engelbart 1962] by augmenting human intellect with knowledge exploration and communication tools. These objectives were adopted in the Alto, 50 of which were distributed to an initial user base of researchers at Stanford and MIT, and by the Star team, who hoped their product would "improve the quality of the user's thinking" [Smith et al. 1982].

When UI technologies for human augmentation (the mouse, bitmapped screen, etc.) were eventually applied to the commercial office market, it was

natural to describe application scenarios in terms of the in-trays, filing cabinets, and wastebaskets that appeared in *Star* and *Pictureworld*. However, an alternative set of tropes can be found in the self-description of the elite research teams who built and used systems like *Dataland* and *NLS*. The users of these systems are made to resemble heroic explorers, adventurers, and wielders of technical power, rather than “office professionals.” In a famous film of Doug Engelbart demonstrating *NLS*, he appears to be piloting [Markoff 2005] a command console rather than sitting at a desk, wearing an audio headset and with controls under each hand. The *Dataland* user sat in a padded chair with controls built into the arms, so he could traverse the imaginary landscape like “a pilot flies an airplane” [Donelson 1978]. The airplane cockpit as a rhetorical site of human-machine symbiosis, synergism, and human augmentation is not unique to 1960’s and 70’s computer science, but has been persistent throughout military-technological fantasies of the 20th century. Powered flight was a pervasive aesthetic of the 1930’s, influencing futurist ideologues like fascist poet Gabriele d’Annunzio, who inspired Mussolini to believe that “a new civili[z]ation would be created by aviation, fusing advanced machines with the steely-eyed heroes who controlled them” [Ballard 2005]. In comparison to these cyborg fantasies of human augmentation or *Star Trek* command chairs, the “desktop” metaphor, portraying mundane office work, might be seen as disappointingly anticlimactic.

My analysis of discussion at the 2003 workshop found both persistent belief in the metaphorical UI as an augmentation fantasy, and disappointment with the actual mundanity of the office desktop. One researcher demonstrated a sophisticated virtual reality system for scientific visualization in which the user acts as a researcher/explorer, flying through solar systems of data to discover uncharted relations. Researchers at the workshop were excited by such visual representations, associating them with the creative mental imagery often reported in scientific discovery. With some effort, an analogy was drawn to the desktop metaphor as a species of scientific visualization where internal abstractions of the computer are “visualized” by a user who must discover them through a process of scientific investigation. As noted previously, this informal discussion would never appear in print—but it does provide an insight into the way in which HCI researchers might generalize from their own intuitive user experience. When their UI designs turn out not to be useful to other people as expected, this was attributed to the user’s failure to “connect” with the metaphor, resulting from alternate cognitive styles or simple aesthetic preference. A common research response is a kind of design abdication wherein the final system incorporates several interface variations (more or less “metaphorical”) from which the user is supposed to select or configure the one that suits him or her best. The computer scientists preferred to ground their own discussion in the common design model underlying these user options, using abstract computational terms, and even explaining the intended interpretation of a particularly complex visual “metaphor” by making an analogy to the *simpler* concept of a relational database system! Novice users were not necessarily expected to ever become competent at understanding the real model. Instead, the interface designer is described as an “expert”

interpreter of the metaphor, and can be contrasted with the permanently “novice” end-user.

3.3 Reifying Creative Experience

Both design researchers and HCI commentators are familiar with metaphor as a strategy for creative design. Schön [1983, 1993] and Visser [2004, p. 86] describe the value of metaphor and analogy in helping all kinds of designers achieve creative intuition and novel inventions. In HCI, Madsen [1994] and Mohnkern [1997] therefore argue, the greatest benefit from metaphor could well be experienced by system designers, rather than system users. Even among philosophers, Fauconnier [1997, p. 171] cites the invention of the Macintosh desktop as a case study of innovation that he attributes to a conceptual blend of the domains of computation and office work (apparently without being aware, at the time, of the detailed history of its development). These analyses correspond to Kay’s intuition about the role of metaphor in his own design process [1990], and he is sceptical that metaphor has brought significant further benefits to users since the “desktop” design was completed.

The design teams at Englebart’s augmentation research center (ARC), Xerox PARC, and Apple placed great emphasis on creativity, reflecting a distinctive Californian culture significantly at odds with traditional corporate attitudes toward technology. In a widely-read feature for Rolling Stone magazine, Stewart Brand [1972] described Kay’s group at PARC as a corporate remnant of 60’s counter-culture, filled with “heads,” “computer bums,” “fanatics,” and “hot rodents” who had “time for screwing around” and were compared to Ken Kesey’s LSD-promoting Merry Pranksters. This culture, along with ideas and individuals, was partly imported from ARC, where experimental LSD use was in fact one element of the experiments in creativity surrounding the NLS project [Markoff 2005]. HCI research today continues to seek generative impetus from the creative arts (e.g., [Candy and Edmonds 2002]), and the appropriation of “literary” metaphor and theories of visual art in the 1970’s might be seen as a precursor to this contemporary pursuit of a technological Renaissance [Leach 2005].

However, it is important to make a clear distinction between the creative work of a computer designer and the creative experience of the computer user. This is made more difficult because the two are intentionally conflated in marketing campaigns (see the next section). As counter-examples in other design domains, consider that a paintbrush supports creative use without being itself a creative product, while an Eames chair, although a creative product, is unlikely to confer creativity on the person sitting in it. Kay’s original goal was not to make a creative design for its own sake, but to make a creative tool for children, which he compared variously to a musical instrument, easel, or printing press [Kay 1972; Kay and Goldberg 1977]. Reaching that goal incidentally required a lot of creative work, although this was a means rather than the end.

As mentioned earlier, visual modes of thought are associated not only with effective communication, but with creativity. Philosophers including Bruner and Johnson describe metaphorical imagery as an essential component of poetic

and of fanciful, as opposed to objective and analytical, modes of thought. Visual representations are open to different readings because, unlike linear texts, the reader avoids the ordered control of an authorial voice [Rotman 1999]. Is the desktop metaphor only communicating a mental model from designer to user (as recommended in design guidelines and textbooks), or are both the designer and user expected to interpret the metaphor creatively? Subsequent to his investigation of Lisa usability, Carroll recommended that users should indeed initiate creative “sense-making” themselves [1997]. But if they are more than passive receivers of communication, their interpretation might not only be unruly, but could draw on irrelevant visual aspects; undesirable “conceptual baggage” [Anderson et al. 1994].

The interplay of creative entitlement between the interface designer and user received substantial attention at the 2003 workshop. Some participants wished to allow vibrant and rich interpretations, while others were concerned that many research prototypes and products were too fanciful, and had been constructed without any clear justification of how they would function as learning aids. Those researchers building computational accounts of metaphor were daunted by the potential variety of interpretations, especially if these were to account for their own creative experiences. Scientific users of the visualization system were described as “appreciating” and being “stimulated” by this rich and complex prototype, while less skilled users of the same system tended to be “overwhelmed.” However, the visual metaphors of educational software had the opposite effect of “dumbing-down” the pedagogic content, forming a literalistic barrier, rather than an invitation to creative exploration.

It is difficult to disentangle researchers’ own experiences of using metaphorical interfaces from their creative experiences of imagining new metaphors. Workshop participants cited Alan Kay, Ted Nelson, and Don Norman, lamenting the loss of “magic” in the over-literal desktop. The desktop suffers from physical constraints, while the abstract world of computation empowers users with creative and magical potential. One contributor suggested that the verbal labels on icons might even be removed in order to allow users more freedom of creative interpretation. Pleasure in creative experience was evident at the workshop as participants developed “riffs” of novel UI concepts from seed metaphors proposed by others, using visual similarities and bizarre images to leap from one concept to another. When compared to such exciting intellectual experiences, the desktop metaphor is not only mundane (an office rather than a laboratory), but moribund. It has been fully integrated into the everyday language of the UI, with no further potency as a generative creative image, one of those everyday dead metaphors that Nietzsche says are “worn by frequent use and have lost all sensuous vigour” [1873]. This is the natural lifecycle of any metaphor, but HCI researchers at the workshop and elsewhere are sometimes disappointed that the desktop seems to mark an end of innovation. Of course, the desktop itself continues to evolve incrementally, and completely new styles of interaction (e.g., iPod, Skype, TiVo, Second Life) are constantly invented. But the desktop still looms as an archetype and obstacle for HCI research. The “Anti-Mac” was a polemical attack on the design principles it embodied [Gentner and Nielson 1996], and the workshop participants wished to mount

similar attacks. For their own purpose, whether or not this benefits users, they sought the theories of metaphor that underlie invention, perhaps Fauconnier's conceptual blends, which were viewed as less deterministic than structural theories of metaphor, thus enabling creative interpretation and expressive freedom for researchers and designers.

3.4 The Business of Reification as Social Control

In light of the preceding discussion, it should be clear that we cannot accept as unproblematic the claim that the desktop metaphor confers creativity and freedom on its users. This claim has been central not only in research rhetoric, but as a pervasive theme in product marketing to establish the connotative entailments of the GUI. Apple's original "1984" advertisement identified the Macintosh with a hammer thrown to destroy a video screen from which Orwell's Big Brother was haranguing an audience of corporate clones. Apple advertising of the 1990's went on to claim Einstein and Gandhi, among other icons of creativity and intellectual freedom, as typical Macintosh users. Microsoft responded with a landscape of blue sky and open horizon in which the user was offered the invitation "where do you want to go today?" As an individual experience, perhaps these machines have achieved some small progress toward the aforementioned cyborg fantasies of human augmentation. Regardless, the people most empowered by the ubiquity of the desktop metaphor have probably been those in the corporations manufacturing the machines.

The transformation of the ideals from libertarian counter-culture into vastly profitable manufactured product has struck many as either paradoxical or profound (e.g., Brand [1995]). During the 1970's, this was already evident in tensions between Californian research laboratories and east coast corporate headquarters, as when Brand's Rolling Stone article jeopardized Alan Kay's research by bringing the latter's existence to the attention of senior Xerox executives [Kay 1996]. Kay's concern with creativity and children was always more likely to find acceptance among hobbyists and the garage ethic of Apple than (for example) the IBM researchers proposing Pictureworld as a user model for "office professionals." Journalists reviewing the Lisa for the small-systems enthusiast market described it as playful and fun, even like a videogame, while, as already noted, real office professionals at IBM (the supposed market) found it frustrating and confusing.

Researchers at the 2003 workshop were still grappling with this conflict 20 years later. They took a moral stand against the imposition of corporate culture, seeing themselves as advocates of users who deserve to be liberated from the constraints of working at a desk, surrounded by filing cabinets and with their daily routine controlled by documents in an in-tray.¹⁵ Ted Nelson was already attacking the "bureaucratic" nature of the desktop metaphor in 1990, not so much because it resembled an office, but because designers were

¹⁵Of course, the in-tray disappeared from the "desktop" metaphor immediately after Star, since the Lisa did not include a standard email system. Furthermore, even the most bureaucratic company in-trays of the 1970's would be unlikely to approach the volume of an average researcher's email today.

so constrained by the need to maintain consistent structural relations between all aspects of the design. But workshop participants were not sympathetic to Nelson because his position challenged the foundation of their own research into new structured approaches. Extended discussion of navigational metaphors in educational multimedia was particularly revealing of the social implications of creativity and control. Poor metaphors were those in which children wasted time “off-task,” instead of being guided to follow prescriptive curricula. What a contrast to Kay’s vision of children using computers for creative exploration! Other discussions at the workshop regularly described the user as a cognitive design resource, the recipient of a mental model constructed by the metaphor designer. There was a discussion of whether the designer or user should “own” the UI metaphor. Whatever the ambitions of HCI researchers, it seems most likely that the thought-space under the umbrella of “metaphor” is owned and controlled, in the same way as any other property of Western society, by the corporations operating our economic and legal systems. When a company designs a “mental model” as an engineering surrogate for an actual person, this company is really *designing the user* to be a better customer for their products.

UI metaphor is a theory constructed in the wake of this commercial success. The inventors of the GUI were certainly motivated by theory, not simply intuition, but their theories were so diverse, and so many of them opposed to the principles eventually formulated for inclusion in textbooks, that we must look elsewhere for an understanding of the theorizing process. Engelbart’s program of human augmentation has been achieved to some extent, but his personal vision is scarcely remembered, with many believing the mouse to be his principal contribution. The original vision passed out of his control, and was at the mercy of those he inspired [Levy 1994; Markoff 2005]. In historical terms, this is typical of commercialization from research prototypes. Historian of science Bruno Latour [1987] claims that the most effective way to spread an idea is to let others transform it. Otto Diesel patented his invention of the Diesel engine, trying to exploit it through commercial channels, but was never able to mobilize the forces to make his prototype work adequately. It was only after the engineering company M.A.N. circumvented Diesel’s patent in designing the first practically viable engine that it became a commercial success. The parallels with Engelbart’s experience are striking, and both Diesel and Engelbart had good reason to be disturbed by the scale of the commercial activity derived from their work in proportion to the personal reward they received.

In Latour’s terms, the construction of science takes place through a network of associations. Those who transform and exploit a scientific advance, like the engineers of M.A.N., enroll themselves as members of a network supporting its truth. These “allies” of the theory include laboratory instruments and experimental subjects, all of whom might bear witness to its truth if called to do so. Latour suggests that the history of science is no more than the history of resources, where competing theories must mobilize larger and more incontrovertible networks of allies. In the case of the desktop metaphor this is dramatically true. The original prototypes were built in the most generously funded centers of the Cold War military-industrial complex. Subsequent commercial explosion of the Macintosh, and then Windows, has established every personal computer

as a “laboratory instrument” of metaphor theory and every user as a witness to its truth. There can scarcely ever have been such a dramatic colonization of the space of human meaning by a technological invention. If we accept the screen metaphor as allowing personal creativity (in either interpretation or use), then the nature of human creativity is being redefined within digital spaces. If we take the pessimistic view of metaphor as a mechanism of technocratic control in which software developers try to “design the user,” then a more paranoid critic might even suspect that the technology research establishment would purposely obscure the nature and provenance of the underlying theories.

The discussions at the 2003 workshop showed evidence of considerable tension around these questions. Participants wished to give users freedom, perhaps through the designer abdicating final control and offering a choice of different metaphors for the user to select and customize. Even maintaining a degree of choice between Windows and Macintosh versions of the desktop might allow users to construct their own experiences. But discussion of practical design cases at the workshop continually led to more conservative formulations. Allowing navigational freedom to users could have the consequence that the user might get lost. In educational software, collections of facts should not be experienced without interpretation, allowing students to draw their own connections, but should be presented in a framework of prescribed relations. Metaphors must be “cohesive” and “complete,” constructed from computationally enumerable relations. Safety for the designer lies in mimicking a nondigital artifact in order that the user’s actions are predictable.¹⁶ The paradigmatic example was to present the UI as if the computer were a book (textbooks, catalogs, and personal calendars were all discussed). Not a book that the user writes (as in Kay’s vision of tools for a creative child), but a book that the user reads, providing a conventionally structured encounter with the world of ideas.

4. THE TOOLS OF HCI AS REIFICATIONS OF METAPHOR

This final part of the article considers contemporary attitudes to metaphor, as revealed in the discussion at the 2003 workshop, and interprets them in light of insights from the sociology of science. In particular, I address the dual nature of HCI as both a research and design discipline. Designers work with tools, and researchers create theories. However, in HCI, both designers and researchers make “products,” whether research prototypes or commercial products. I consider the implications of this relationship from several different perspectives, each arising from my analysis of the workshop discussion. My main interest is in whether we can draw useful lessons for the future of HCI, and what kind of metaphorical endeavors may or may not be sensible for us to undertake.

4.1 Design Case Studies as Reified Theories

Whatever theories of metaphor are espoused by HCI researchers, it is through the work of design that these theories become embedded in products, tested in

¹⁶Described by Pirhonen [2005] as the “simulation” rather than “stimulation” approach to metaphor.

the market, and (hopefully) adopted by a large base of users. This situation is not comfortable for scientists who aspire to truths that are permanent, rather than the historical contingencies of solving an actual engineering problem [Vincenti 1990]. One response to the historical contingency of metaphor research is to recast human and social aspects as analytic computational formulations that are more likely to endure. An alternative strategy is to retrospectively interpret scientific research as motivated by, and hence answering, design questions. One workshop participant recalled Gentner and Stevens' work on mental models as an investigation of the issues raised by Norman's popular design text *The Psychology of Everyday Things* [1988], although Norman's book was not in fact published until five years later. Both of these strategies are uncommon in contemporary HCI. Most HCI researchers at the metaphor workshop and elsewhere aim to influence product design by influencing designers.

The most constant theme arising from my analysis of the workshop transcript was the question of how metaphor research should be situated relative to professional UI design. For these researchers, professional designers act as gatekeepers who determine the fate of metaphor research through their influence over future products, and because they possess the material and craft resources that can transform theories into persuasive artifacts. Researchers described themselves as interpreters of interdisciplinary authority (from literature, philosophy, and psychology) attempting to "package" the results as prescriptive design methodologies or critical evaluation methods. However, they find themselves more often in the role of evangelist or even supplicant, rather than interpreter or oracle. One contributor complained that designers "pay lip-service to metaphor theory," and another that they "never read books." There was much talk of mounting a strategic campaign against design commentators such as Norman, who was accused of betraying researchers by denying the significance of metaphor. But there was little agreement on what strategy would be most effective in this campaign. It was suggested on the one hand that designers cannot follow abstract critical discussion, so must be provided with concrete design elements. On the other hand, scientists were seen as losing the strategic high ground when designers follow facile design formulae, in which case the solution is further intellectualization of designers' craft work. Coyne suggests that "[a]n emphasis on metaphor also represents a liberal attitude to design. Through such emphasis, design is cast largely in terms of devising appropriate metaphors rather than solving a problem through theoretical analysis" [Coyne 1995, p. 250]. When metaphor design is seen in this light, perhaps a concern with developing even more theory is very much missing the point!

There is a further tension, in a research field that has valorized creative generativity in metaphor use, that the creative work of design cannot easily be reduced to the status of a research obstacle. Kay's original ambition to develop new creative tools was accompanied by great respect for the creative professions. He hoped that his own software team would see themselves as craftsmen and "artisans" [Brand 1972], and has said that Dataland was so influential because it was uniquely beautiful as the only UI research prototype of the 1970's that had employed a professional designer [Kay 2002]. Commitment to professional visual design pervaded Apple [Mountford 1990], and some reflections on

the period see metaphor not as a cognitive or philosophical principle, but simply as the visual design language associated with a particular class of product (e.g., the spreadsheet, as described by Fylstra [2004]). A recent publication by Carroll [Go and Carroll 2004] seems to have moved toward this usage, describing UI metaphor simply as an aggregate of visual design elements, as opposed to the systematic cognitive structures described in his earlier work.

At the 2003 workshop, participants aspired to creative design of new metaphors. There were several attempts at inventive brainstorming, in the hope that deep understanding of metaphor principles could lead directly to design innovation. In practice, these creative efforts quickly subsided into critical comparative discussion of innovative research prototypes created by others in the past, and then to critiques of commercial products viewed as deficient applications of design metaphor. Idiosyncratic products were described as having failed, not because metaphor had been applied inappropriately, but because the designers did not meet the requisite professional standard, being ignorant of basic graphic design or usability principles. In much of the discussion, “metaphor” became a kind of sheriff’s badge, entitling the theoretically informed research critic to make observations about (for example) the legibility of mobile phone screens and the failure of young designers to account for the decreasing dexterity and visual acuity of sadly aging academics. The one UI that escaped such critique was, ironically, the desktop metaphor. Microsoft Windows (and Google, and TiVo) were cited in research arguments because of their commercial leadership, but it seems that the ubiquitous Windows desktop, as the paradigm and canonical example of successful UI metaphor, cannot be directly attacked without undermining the relevance of metaphor research itself.

The complex relationship in HCI between theoretical descriptions and designed products can be better understood by comparison to the relationship between theory and practice in the medical profession, and here I draw on Susan Leigh Star’s description of the origins of neuroscience [1989]. Any particular UI design is like a clinical case study, which although it might illustrate theory, cannot be a straightforward exemplification of theory due to the individual circumstances of the patient. Scientific debate tends to focus on abstract ideal types, whether realized as brain maps (in neuroscience) or cognitive models (in HCI), whereas clinicians and designers must focus on solving a specific problem. Nonetheless, in the medical profession, as in HCI, many innovative practitioners are also researchers. Theoretical developments are thus linked to professional reforms that lend *inertia* to the scientific process. As a particular clinical practice is adopted, the accumulation of case studies adds *momentum*, and the necessities of persuasion—whether other practitioners or competing theories—lead to *reification* of the abstract ideal types in actual case studies. In HCI these forces lead to the demonstration culture (notoriously in the advice given to research students at the MIT Media Lab that they must “demo or die”). The demo culture of research presumes that a realized design embodies metaphorical and other theories. Every clinical case study, as Star says, has so many specifics of the context of use, problems overcome, and visual details that it is convincing in itself; there is little further need to add or defend theory. Kay reviews his research achievements in the pragmatic formula that “the best way

to predict the future is to invent it.” The reification of metaphor, and of many theories in HCI, has been carried out through the exhibition of design work. In Latour’s terms, designed software products serve as theories that have been made into instruments. Every user of Windows has thus been enlisted in confirming the cognitive status of the UI metaphor by performing his or her own translation and interpretation of this particular design.

4.2 Critical Assessment of Novel Metaphors

As explained in the introduction to this article, I have focused on the desktop as the historical origin and paradigmatic example of interface metaphor. This has been motivated by concision and clarity of argument, but most of the issues that I have raised apply equally to other established and proposed metaphors. Potential examples might include projects developing new UI metaphors for television [Nonogaki and Ueda 1991], information retrieval [Dieberger and Frank 1998], or tangible interaction [Svanaes and Verplank 2000]. Large projects such as these deserve a more detailed critique than there is space for in this article. However, it is interesting to consider the foundations of some novel metaphors discussed during the 2003 workshop because these can be compared to the application case studies presented in Chapter 6 of de Souza’s superb book, *The Semiotic Engineering of Human-Computer Interaction* [2005]. De Souza discusses alternative conceptual metaphors for computer-mediated communication, such as *virtual environments* in which people can move around and meet each other, or *communication centers* staffed by (simulated) agents who guide users toward their goals. Although she justifies the basis for conceptual metaphor in the work of Lakoff (as well as Halasz and Moran, as the first interpreters of Lakoff for HCI), her main focus is on the communicative function of the resulting metaphors, rather than the design philosophies they reveal.

The virtual environment metaphor, presenting a simulated physical world, can be compared to a repeated slogan of the Xerox Star developers that “display becomes reality.” Cultural commentators [Levy 1994; Rheingold 2000] have naturally seized on this provocative claim. De Souza describes it as a reification of computational concepts into physical space. However, this process is an inversion of Piaget and Bruner’s conception of symbolic learning arising from concrete experience. As an ambition to empower the user through new experience, it simultaneously extends the supposed power of the computer from the mental realm into the physical world. The 2-D graphical interface no longer strikes us with such force, but at the 2003 workshop, many participants remained concerned with three dimensions. They described the “desktop” depicted on the screen as if it were an actual 3-D space (despite the commercial failure of systems such as Bob and Magic Cap that did this literally, and the lack of proof that 3-D metaphor prototypes provide measurable benefits [Cockburn and McKenzie 2002]). Participants wished to exploit the latest rendering technologies developed for PC videogames, much as Kay exploited contemporary rendering technologies of 30 years ago (bitmapped display and clipping windows). One participant described a metaphor for users immersed in a 3-D virtual reality “CAVE”—drawing on William Gibson’s cyberspace, the science fiction trope that

has now replaced aviation as a fantasy of technologically-augmented individual power worthy of Foucault.

Alternatively, the metaphor of the communications center involves the computer acting as a social agent, perhaps even a conversational partner. Workshop discussion of this strategy regularly strayed into issues of artificial intelligence and references to the Turing test. De Souza notes this problem in passing as an anthropomorphizing of the computer, but her main focus is on communication between the system designer and user, where the system acts as a “designer’s deputy” that can only offer “one-shot” communication, instead of conversation. Nevertheless, some workshop participants described their ambition as the creation of a computer that could become a conversational partner in maintaining a metaphor, while others rejected this “linguistic” interpretation, focusing on the computer as a machine that we operate, not converse with. This conflict was not resolved. Models of artificial life were invoked where the computer would even participate in building new metaphors. Researchers speculated about human-machine conversations in future voice interfaces, and the need to maintain a dynamic software model of the user’s understanding; a theory of mind for the otherwise autistic computer. These informal speculative debates have deep roots. Natural language is slippery and polysemous, and metaphor use among humans is always ambiguous, based on social context. In the event that a computer appears to be a social agent, Collins and Kusch [1998] point out that this arises from willing users unconsciously applying their own social competence to *repair* social errors of the machine, and *attributing* the resulting apparent competence to artificial intelligence, rather than to the joint intents of the user and designer as mediated by the machine.

There is ample opportunity for the development of more sophisticated critical theory in HCI. De Souza offers a pragmatic engineering framework for design work, but this could usefully be supplemented by further theoretical perspectives from philosophers such as Foucault, or Collins and Kusch. Other product design disciplines have been more advanced than HCI in the application of critical and literary theory to product design [Richardson 1993; Crilly 2005], perhaps because they are not inclined to rely so heavily on computational accounts of their users. For readers in HCI, the book that I would recommend for further reading on this topic is Richard Coyne’s *Designing Information Technology in the Postmodern Age* [1995].

4.3 Implications for Design Practice

“[W]e don’t have a good understanding of why certain core interface techniques are successful. GUIs are central to the enterprise of HCI, and although we have successful guidelines for building them [Shneiderman 1982], we lack theoretical understanding of why they are successful.”

[Whittaker et al. 2000]

There have been many attempts over the past 20 years to regularize the theory of interface metaphor. A common theme is that the term “metaphor” should itself be taken as a metaphor. This is both a licence to broad interpretation, and a convenient simplification of a complex phenomenon, as when a physicist

talks metaphorically about gas molecules as though they were billiard balls. The simplified metaphor of “metaphor” in HCI stands for a complex aggregate of design concepts borrowed from technical literature in cognitive science, education, philosophy, and the design of computer displays and control interfaces. The technical heritage has been reified in designed products, incorporating the many incremental inventions of object-oriented direct manipulation. However, the technical literature has been supplemented by a large *nontechnical* literature [Strauss and Corbin 1990] of commentary and reflection, and this nontechnical literature is primarily focused on modifying the practice of design. In the extreme, this extends to prescriptions of design methods that might routinely be applied as a standardized tool of interface designers.

When we review the nontechnical commentary, we find that it has not converged on any single reified tool, but rather to a collection of observations about what metaphor is *not*. It has been said that metaphor is not spatial [Johnson 1997] because attempts to construct interfaces that depict physical space have failed to deliver their promises, whether they make it to market (Magic Cap and Bob) or not (Apple’s “Hot Sauce” prototype by Guha, and Microsoft’s “Task Gallery” by Robertson et al. [2000]). Others say that metaphor is not a systematic relation to the “real” world, because the real world is inadequate to express computational operations [Johnson 1985]. Prototypes based on more depictive “real-world” pictorial metaphorical interfaces can seem perverse in their application of metaphor, as when Kosara et al. [2002] portray a medical information system as a running track near which road signs and traffic lights appear. Metaphor seems not to depend on any physical substance, offering only illusion [Kay 1990], a postmodern presentation of surface without depth [Turkle 1999], or a medium in which we explore our own will [Swigart 1990]. This article opened with Nelson’s [1990] complaint that the systematic pursuit of metaphorical consistency is a bureaucratic dead weight on innovation. Finally, Weiser’s [1994] vision of ubiquitous and invisible computing implies that metaphor should not really be anything at all. UI metaphors draw attention to themselves, rather than fading into the unremarkable background of the digitally augmented world that he proposes.

This critical commentary warns us of dangers in applying metaphor for the wrong reasons, but provides few clues about what the right reasons might be. Participants in the 2003 workshop attended because they also wished to move beyond entrenched critical stances, and to make contributions to future design practice. Their discussions circled around three themes (although the formulation of these themes has arisen from my analysis, as opposed to being identified explicitly at the time). First, researchers were motivated by an ideal of completely natural, intuitive interaction. Second, they hoped that it might be possible to prescribe a methodology for metaphorical design. Third, they wanted to create user experiences that are rich, creative, and affectively satisfying.

With regard to the first of these aspirations, to achieve intuitive interaction, there is little evidence to encourage researchers. Novel metaphorical UIs, despite their popularity, have seldom been natural or intuitive. Computer experts admired the conceptual innovations of the Star and Lisa, but regular users found them confusing and challenging, especially if they already had

some experience with conventional interfaces. Design innovators appropriated a variety of concepts from cognitive psychology as scientific warrants for the “natural” perceptual and mental benefits of UI metaphor, but these have never been more than partial explanations. Young researchers today who feel that the desktop is “intuitive” are principally beneficiaries of a convention that has been ubiquitous since their own childhoods, making it difficult for them to imagine what it would be like to be equally enculturated to a different interaction paradigm. The apparently intuitive nature of today’s desktop, as well as its capacity to disappoint design researchers, is elegantly expressed by philosopher Richard Rorty:

What we call common sense—the body of widely accepted truths—is, just as Heidegger and Nabokov thought, a collection of dead metaphor. Truths are the skeletons which remain after the capacity to arouse the senses—to cause tingles—has been rubbed off by familiarity and long usage. After the scales are rubbed off a butterfly’s wing, you have transparency, but not beauty—formal structure without sensuous content. Once the freshness wears off the metaphor, you have plain, literal, transparent language—the sort of language which is ascribed not to any particular person but to ‘common sense’ or ‘reason’ or ‘intuition’

[Rorty 1989, p. 152]

The second aspiration of the 2003 workshop was the search for a methodology of metaphorical design. Regardless, it seemed that the most successful metaphors were not necessarily the product of a systematic design process, but rather a single discovery. One participant argued that the most effective recent application of metaphor was the “shopping basket” on online shopping sites. This functions by direct analogy to a nearly identical domain, with no need for theoretical justification or a detailed theory of educating the user. If the design of the shopping basket metaphor resulted from a single insight, with implementation details following straightforwardly, what advice would help designers to find and apply such metaphors in the future?

The most substantial theoretical support for metaphor in HCI now comes from Lakoff and Johnson’s cognitive semantics, which argues that any human understanding of abstractions, whether computational or in other domains, will be based on embodied metaphorical images.¹⁷ However, this universal semantic principle would apply to all UIs, whether or not the designer intended them to be “metaphorical.” Furthermore, the designer is just as much an embodied thinker as the user, so there is no perspective from which any design could be nonmetaphorical. From this perspective, the best advice to a designer would be to follow his or her common sense, as it is common sense rather than analytic theory that will correspond most closely to the user’s own natural understanding. As an example applied to the shopping basket, my expert advice might be that members of the set of items to be purchased should be depicted on the screen in an area that is circumscribed by a visual boundary, thereby indicating containment. This spatial “inside” relation would correspond to our

¹⁷Cognitive semantics is still only one possible epistemological stance. As noted by Coyne [1995, p. 275], Heidegger argues that there are more basic forms of experience than embodiment, and that Lakoff and Johnson’s account of metaphoric schemas might be deconstructed from this perspective.

image-based semantic understanding of the “inside” of a mathematical set, and also to the physical experience of a basket having an “inside.” One researcher has constructed a systematic theory of such commonsense properties [Treglown 1999], drawing on an essay by a philosopher applying cognitive semantics to the Macintosh in an equally commonsense manner [Rohrer 1995]. However, it is unlikely that designers would ever attempt a different solution, and if they did, it would soon receive commonsense criticism from colleagues, reviewers, or trial users. This limits the audience for such theory-based design advice.

In preceding sections, I have described a variety of nontechnical factors that will probably continue to motivate proposals of new systematic design methods based on metaphor. Design methodologies can be used to enforce objective perspectives, maintain the professional status of research over craft-based design, and to structure mechanisms of corporate control over the context of use. These operational accounts of theory-based design might be founded in philosophy [Treglown 1999], applied linguistics [de Souza 2005], cognitive anthropology [Anderson et al. 1994], rhetoric [Marcus 1994], or indeed in any one of the disciplines that have constructed rich accounts of metaphor. HCI researchers wishing to propose a new methodology of their own could start with one of the chapters in Ortony’s *Metaphor and Thought* [1993]. There are still a couple of dozen available to be used. Design methodologies will continue to be required in prescriptive textbooks of design practice (e.g., Beyer and Holtzblatt [1997, pp. 269–271]), and to provide theoretical justification when researchers invent novel UI techniques (e.g., Yousef [2001]), so there should be sufficient demand to match this ample supply.

Extrapolating from previous methodology research, I would expect that proposed design methodologies will continue to be distinguished by an implicit or explicit computational theory of mind. Research will be characterized by building some simulation of user cognition, structured in accordance with whatever metaphor theory has been chosen. Designers will be offered analytic tools with which to deconstruct their design concepts into structural components and objective relations. The advice offered to designers will be primarily critical rather than generative, so will not provide any significant support for the primary (creative) contribution that a designer is expected to make to a development team. The 2003 workshop included substantial discussion of this kind of methodological approach, but framed by a feeling of frustration about what could be achieved by it. There was a feeling that the resulting critiques focused on trivial design details, and did not address the potential of metaphor in a holistic way. Current design practice was itself criticized for relying on guidelines and idioms that, in spite of having some metaphorical content individually, are incoherent overall. Participants were uncomfortable that although they could construct critiques of complex products like Microsoft Office, structural analysis could contribute little understanding to a new generation of products that, like Google, succeed through simplicity of interface, rather than structure. This kind of simplicity was described as trivial, suitable for children, and by implication, only novice adults. Complex and expensive research prototypes employing exotic technologies such as virtual reality are no more likely to become widespread than the flight chair of SDMS Dataland, so the eventual outcomes of such research are

the details, not the vision, as in the case of NLS, a complex and ambitious project now remembered for the invention of the mouse.

The third aspiration of the workshop participants was the desire for richly affective user experience. Nonetheless, the discussion showed suspicion of the more designerly aspects of metaphorical interface. Despite the central role that visual design played in the success of the Macintosh, visual appeal was seen as opposed to the deep structural relations that should characterize true metaphorical interfaces. Designers were accused of concentrating on the surface, choosing metaphorically arbitrary images simply because they were attractive. In contrast to Kay's great respect for the creative design professions, contemporary researchers appeared to mistrust those responsible for the visual appearance of the interface. Graphic designers were described as being unaware of the importance of mental models, and of naively confusing the visual image with the resultant mental image. Designers were accused of making interfaces that are "cute," engaging the user's visual attention in a frivolous manner that is wasteful of mental resources. Even good visual designs can be a source of discomfort to metaphor researchers because an account of design quality relies on subjective aesthetic judgments that cannot easily be defined in terms of the "computer science world."

After 10 years studying UI metaphor, including analysis of its historical and philosophical origins, as well as contemporary research trends, my own opinion is that we must pay more serious attention to the creative richness and affect that is valued in professional design.¹⁸ Reductionist accounts of the user cannot make a useful contribution toward this end. Creative metaphor may continue to be a useful generative tool for the designer, but an even more significant change would be to recognize its value as a creative experience for the user, as noted in recent work by Pirhonen [2001]. Literature and the arts do not aspire to achieve a predetermined response from the "user" of the artwork, but to evoke rich experience. Philosopher of language David Donaldson says that "[m]etaphor is the dreamwork of language and, like all dreamwork, its interpretation reflects as much on the interpreter as on the originator" [Donaldson 1984, p. 245]. I believe that the attributes of good UI design should be described in qualitative and aesthetic terms, applying approaches from critical theory, instead of empirical studies or automated analysis and critique. In my own research I have therefore shifted my focus to breaking, rather than making, interactive paradigms. In transferring the results of such research to professional design practice, my ambition would be to establish productive new design lexicons for future products. In fact, this corresponds very well to the commonplace use of the term "metaphor" among professional designers, which seems to correspond to the terms "genre" and "style" in other artistic fields, as opposed to anything discussed at the 2003 workshop. It is a waste of time to legislate against language change. We researchers should accept this more approximate and ambiguous use of the term, responding to it with relevant

¹⁸This applies not only to the user interface, but also to the design of software architectures, as noted by Coyne [1995, p. 295] and in the more applied work of Noble et al. [2001] or Imaz and Benyon [2007].

commentary and experimental innovations. There will continue to be opportunities to operationalize theories of metaphor within a design process, but this is likely to be achieved by continued critical reappraisals of designed products, deconstructing the claims of “metaphor,” and constructing new descriptions as to why these designs are successful.

5. CONCLUSIONS—FOR HCI AND FOR CHI

Innovations for graphical interaction with computers started to appear in the early 1960’s, building technical and intellectual momentum through the 1970’s. Concern with the human consequences of technology is associated with the *Zeitgeist* of those decades [Winner 1986], and the UI research and development teams at Xerox, IBM, and Apple were motivated not only by increasing product sales, but also by democratic access to technology, as opposed to fantasies of the augmented individual within Cold War era technosystems. This historical change had already played out by the time of the 1982 conference in Gaithersburg that preceded the establishment of the ACM *CHI* series. Although described as a “human factors” conference, very few articles were devoted to traditional ergonomics; the majority proposed either cognitive user models or design methodologies.¹⁹ Smith and Irby [1998] later claimed that the work of the Star team had motivated the formation of ACM *SIGCHI*, although the actual development of CHI out of the field of human factors was far more complex [Grudin 2005]. The Star project certainly impressed the research community, just as it did the hobbyist and “small-systems” community of *Byte* readers, but my analysis suggests that the theoretical and strategic attitudes of HCI have run in parallel to design innovations, rather than having a direct relationship. It is in analytic accounts, theoretical explanations, and the writing of HCI textbooks that a structured theory of UI metaphor became prevalent.

Workshop participants in 2003 were uncomfortable with the legacy of HCI research into metaphor. They personally believed that metaphor was one of the few certainties in HCI, describing it as a “golden rule” and “widely acknowledged” to be essential. But this position was under siege. Well-known critics of metaphor in the HCI community were portrayed as enemies who believe that metaphor is “fundamentally pointless,” and who are intent on “leaving the world devoid of metaphor.” The embattled community of metaphor researchers at this British workshop and a previous one in Greece [Rachovides and Swiderski 2001] wished to “reclaim” metaphor as an object of academic enquiry, obtaining the right to build their own critical accounts by inspecting the ecology of designed artifacts, and unpicking those aspects of the design that correspond to theories of metaphor. However, a critical strategy of this kind is not generally welcome within any technology discipline. The current climate of computer science research in the UK is that of intellectual pursuit of “grand challenges” in technocentric computer science. This is accompanied by huge investment in

¹⁹Of the 74 papers presented in Gaithersburg, only six seem recognizable as belonging to the field of ergonomic human factors. There were eight descriptions of new interaction technologies, and ten empirical studies of technology in use, but the majority were devoted either to cognitive models of the user (28 papers) or design methodologies (12 papers).

“e-Science” grid computing, where even researchers must submit to becoming components of engineering systems, their function measured and monitored via a national research assessment exercise. Despite being computer scientists themselves, the workshop participants were worried that these trends in their own academic profession were symptoms of a systematic devaluing of human perspectives in technology design, and hoped that metaphor might provide a bridge via which they could become interpreters of the human insights from other academic disciplines.

The workshop treated other disciplines as supplying an academic research-warrant that would provide external authority when counterclaims regarding academic quality in “practice-based” design research must be defended. This is a particularly technocentric view of academic life. A more nuanced view of intellectual contributions to technology research comes from Weiser [1994], who proposes that the design of “invisible” ubiquitous computing systems must be approached with the assistance of the humanities and social sciences because these disciplines “expose the otherwise invisible.” This has been the motivation of my own work on the project. Myers’ [1998] history of the development of user interaction hardware invited a complementary future history of the “human side” of the UI. My goal here has not been purely historical, but rather to draw attention to the fabric we have woven from strands that include the historical, documented visions of the pioneers, the reified practices of commercial design, and the relations within contemporary technology research. Although myself a technologist, I have had to adopt the methods of social science in order to do this job adequately. The reasons for doing so have been divided by Edge [1994] into either educating others in the techniques necessary for smooth operation of the status quo or understanding the institutional and social forces that drive our own research. At present, I think I have made more progress toward the latter, perhaps at the expense of the former. The driving force of reification is a response to an ill-structured problem [Star 1989]. A reified theory is more easily described, not too complex, and more likely to stay fixed in a research community. Among the broad community of HCI research and professional practice, “metaphor” has reached this state. But in so arriving, many original concerns, still profound and central problems, have been neglected. Whether or not we continue to use the word metaphor in this sense, we must not lose sight of the questions that it once raised. Conceptual change requires the adoption of new languages [Rorty 1989] and new languages are assembled out of metaphor [Hesse 1980]. As researchers, we must also be metaphor users, in order to invigorate the professional designs and user experiences of the future.

ACKNOWLEDGMENTS

I am grateful to Bill Buxton, Jack Carroll, Nathan Crilly, Adele Goldberg, Thomas Green, Jonathan Grudin, Peter Jones, Ken Kahn, Wendy Kellogg, Martin Kusch, Matthias Müller-Prove, William Newman, James Noble, Antti Pirhonen, Debbie Reese, Tim Rohrer, Mary Beth Rosson, Larry Tesler, Bill Verplank, Alan Wall, Richard Young and several anonymous reviewers for their feedback and contributions to this work. Participants in the 2003 *Metaphor in HCI Workshop* were Tom Addis, Fran Bacon, David Benyon, David Billinge,

Sue Fenley, Silvia Gilotta, Amy Hogan, William Hudson, Olga Anna Kuchar, Mark Lee, Dorothy Rachovides, Mark Treglown, Andrew Vande Moere, Rosita Venousiou and Alan Wallington. This article is dedicated in memory of Mark Treglown.

REFERENCES

- ANDERSON, B., SMYTH, M., KNOTT, R. P., BERGAN, M., BERGAN, J., AND ALTY, J. L. 1994. Minimising conceptual baggage: Making choices about metaphor. In *People and Computers IX (Proceedings of HCI'94)*, G. Cockton et al., eds. 179–194.
- APPLE COMPUTER, INC. 1992. *Macintosh Human Interface Guidelines*. Addison Wesley, Reading, MA.
- AUSUBEL, D. P. 1960. The use of advance organizers in the learning and retention of meaningful verbal material. *J. Educ. Psychol.* 51, 5, 257–262.
- AUSUBEL, D. P. 1968. *Educational Psychology: A Cognitive View*. Holt, Rinehart, and Winston, New York.
- BÆRENTSEN, K. B. AND TRETTEVIK, J. 2002. An activity theory approach to affordance. In *Proceedings of the NordiCHI Conference*, 51–59.
- BALLARD, J. G. 2005. Up with the celestial helmsman. Review of R. Wohl “The spectacle of flight: Aviation and the Western imagination 1920–1950”. *Guardian Rev.*, (May 7).
- BARBOSA, A. D. J., PALANQUE, P., NAVARRE, D., AND BASTIDE, R. 2005. Direct manipulation and conversational interfaces for supporting model-based end-user development. In *End-User Development*. H. Lieberman et al., eds. Kluwer, Dordrecht, the Netherlands.
- BEWLEY, W. L., ROBERTS, T. L., SCHROIT, D., AND VERPLANK, W. L. 1983. Human factors testing in the design of Xerox’s 8010 “Star” office workstation. In *Proceedings of the CHI Conference*, 72–77.
- BEYER, H. AND HOLTZBLATT, K. 1997. *Contextual Design: A Customer-Centered Approach to Systems Design*. Morgan Kaufmann, San Francisco, CA.
- BLACK, M. 1962. *Models and Metaphors*. Cornell University Press, Ithaca, NY.
- BLACK, M. 1993. More about metaphor. In *Metaphor and Thought*, 2nd ed. A. Ortony, ed. Cambridge University Press, Cambridge, UK. 19–41.
- BLACKWELL, A. F. 1996. Metacognitive theories of visual programming: What do we think we are doing? In *Proceedings of the IEEE Symposium on Visual Languages*, 240–246.
- BLACKWELL, A. F. 2001. Pictorial representation and metaphor in visual language design. *J. Visual Lang. Comput.* 12, 3, 223–252.
- BLACKWELL, A. F., ED. 2001. *Thinking with Diagrams*. Kluwer Academic, Hingham, MA.
- BRAND, S. 1972. Spacewar: Fanatic life and symbolic death among the computer bums. *Rolling Stone* (Dec. 7).
- BRAND, S. 1995. We owe it all to the hippies. *Time Mag.* 145, 12.
- BRUNER, J. S. 1966. *Toward a Theory of Instruction*. Harvard University Press, Cambridge, MA.
- BUSH, V. 1945. As we may think. *Atlantic Monthly* 76, 101–108.
- BUXTON, W. 2005. Interaction at Lincoln laboratory in the 1960’s: Looking forward—Looking back. In *Proceedings of the CHI Conference*. 1162–1167.
- CANDY, L. AND EDMONDS, E. A. 2002. *Explorations in Art and Technology*. Springer Verlag, New York.
- CARD, S. K. AND MORAN, T. P. 1986. User technology: From pointing to pondering. In *Proceedings of the ACM Conference on the History of Personal Workstations*, 183–198.
- CARROLL, J. M. 1997. *Minimalism Beyond the Nurnberg Funnel*. MIT Press, Cambridge, MA.
- CARROLL, J. M., MACK, R. L., AND KELLOGG, W. A. 1988. Interface metaphors and user interface design. In *Handbook of Human-Computer Interaction*. M. Helander, ed. North-Holland, Amsterdam. 67–85.
- CARROLL, J. M. AND MAZUR, S. A. 1986. LisaLearning. *Computer* 19, 11, 35–49.
- CHEN, P. 2002. Entity-Relationship modeling: Historical events, future trends and lessons learned. In *Software Pioneers*. M. Broy and E. Denert, eds. Springer Verlag, Berlin. 296–309.
- CHILDPRESS, D. S. 1989. Hector Kay, Alan Kay and the future of prosthetics and orthotics Hector Kay memorial lecture. *J. Ass. Children’s Prosthetic-Orthotic Clinics* 24, 2, 40.

- COCKBURN, A. AND MCKENZIE, B. 2002. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In *Proceedings of the CHI Conference*, 203–210.
- COLLINS, A. AND GENTNER, D. 1987. *Cultural Models in Language and Thought*. D. Holland and N. Quinn, eds. Cambridge University Press, Cambridge, MA. 243–265.
- COLLINS, H. M. 1994. Science studies and machine intelligence. In *Handbook of Science and Technology Studies*. S. Jasanoff et al., eds. Sage, Thousand Oaks, CA. 286–301.
- COLLINS, H. AND KUSCH, M. 1998. *The Shape of Actions: What Humans and Machines Can Do*. MIT Press, Cambridge, MA.
- COOPER, A. 1995. *About Face: The Essentials of User Interface Design*. IDG Books, Foster City, CA.
- COYNE, R. 1995. *Designing Information Technology in the Postmodern Age*. MIT Press, Cambridge, MA.
- CRILLY, N. 2005. Product aesthetics: Representing designer intent and consumer response. Unpublished PhD Thesis, University of Cambridge.
- DAVIDSON, D. 1984. *Inquiries Into Truth and Interpretation*. Oxford University Press, Oxford, UK.
- DELEUZE, G. AND GUATTARI, F. 1980. *Mille Plateaux*. Minuit, Paris; tr. B. Massumi. 1987. *A Thousand Plateaus*. University of Minnesota Press, Minneapolis, MN.
- DE SOUZA, C. S. 2005. *The Semiotic Engineering of Human-Computer Interaction*. MIT Press, Cambridge, MA.
- DE SOUZA, C. S. AND BARBOSA, S. D. J. 2006. A semiotic framing for end-user development. In *End-User Development*. H. Lieberman et al., Eds. Kluwer, Dordrecht, the Netherlands. 401–426.
- DE SOUZA, C. S., BARBOSA, S. D. J., AND PRATES, R. O. 2001. A semiotic engineering approach to user interface design. *Knowl. Based Syst.* 14, 8, 461–465.
- DIEBERGER, A. AND FRANK, A. U. 1998. A city metaphor to support navigation in complex information spaces. *J. Visual Lang. Comput.* 9, 597–622.
- DIX, A., FINLAY, J., ABOARD, G. L., AND BEALE, R. 1998. *Human Computer Interaction*, 2nd ed. Prentice Hall, Hemel Hempstead, UK.
- DONELSON, W. C. 1978. Spatial management of information. In *Proceedings of the SIGGRAPH Conference*, 203–209.
- EDGE, D. 1994. Reinventing the wheel. In *Handbook of Science and Technology Studies*. S. Jasanoff et al., eds. Sage, Thousand Oaks, CA. 3–23.
- ENGELBART, D. C. 1962. Letter to Vannevar Bush, dated May 24 1962. <http://sloan.stanford.edu/mousesite/EngelbartArticles/LetterToVBush.html>.
- ERICKSON, T. D. 1990. Working with interface metaphors. In *The Art of Human-Computer Interface Design*. B. Laurel, ed. Addison Wesley, Reading, MA. 65–73.
- FAUCONNIER, G. 1997. *Mappings in Thought and Language*. Cambridge University Press, Cambridge, UK.
- FAULKNER, C. 1998. *The Essence of Human-Computer Interaction*. Prentice Hall, Hemel Hempstead, UK.
- FOSS, D. J., ROSSON, M. B., AND SMITH, P. L. 1982. Reducing manual labour: An experimental analysis of learning aids for a text editor. In *Proceedings of the CHI Conference*, 332–336.
- FYLSTRA, D. 2004. VisiCalc: Design tradeoffs in the first electronic spreadsheet. After dinner speech at the 5th Annual Conference of the European Spreadsheet Risks Interest Group (*EuSpRIG*). Klagenfurt, Austria (Jul. 15–16).
- GALITZ, W. O. 1997. *The Essential Guide to User Interface Design*. Wiley, New York.
- GENTNER, D. AND GENTNER, D. R. 1983. Flowing waters or teeming crowds: Mental models of electricity. In *Mental Models*. D. Gentner and A. L. Stevens, eds. Lawrence Erlbaum Associates, Hillsdale, NJ. 99–130.
- GENTNER, D. AND NIELSON, J. 1996. The anti-Mac interface. *Commun. ACM* 39, 8, 70–82.
- GENTNER, D. R. AND GRUDIN, J. 1996. Design models for computer-human interfaces. *Computer* 29, 6, 28–35.
- GIBSON, J. J. 1979. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA.
- GLASGOW, J., NARAYANAN, N. H., AND CHANDRASEKARAN, B., EDS. 1995. *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. AAAI Press, Menlo Park, CA.

- GO, K. AND CARROLL, J. M. 2004. The blind men and the elephant: Views of scenario-based system design. *Interactions*, Nov./Dec., 45–51.
- GOLDBERG, A. AND ROBSON, D. 1979. A metaphor for user interface design. In *Proceedings of the 12th Hawaii International Conference on Systems Sciences*, 148–157.
- GRUDIN, J. 2005. Three faces of human-computer interaction. *IEEE Annals Hist. Comput.* 27, 4, 46–62.
- HACKOS, J. T. AND REDISH, J. C. 1998. *User and Task Analysis for Interface Design*. Wiley, New York.
- HALASZ, F. AND MORAN, T. P. 1982. Analogy considered harmful. In *Proceedings of the Conference on Human Factors in Computer Systems*, 383–386.
- HEROT, C. F., CARLING, R., FRIEDEL, M., AND KRAMLICH, D. 1980. A prototype spatial data management system. In *Proceedings of the SIGGRAPH Conference*, 63–70.
- HESSE, M. B. 1966. *Models and Analogies in Science*. University of Notre Dame Press, Notre Dame, IN.
- HESSE, M. B. 1980. *Revolutions and Reconstructions in the Philosophy of Science*. University of Indiana Press, Bloomington, IN.
- HILL, S. 1995. *A Practical Introduction to the Human Computer Interface in a Semester*. DP Publications, London.
- HOFFMAN, R. R., COCHRAN, E. L., AND NEAD, J. M. 1990. Cognitive metaphors in experimental psychology. In *Metaphors in the History of Psychology*, D. E. Leary, ed. Cambridge University Press, New York. 173–229.
- HORTON, W. 1994. *The Icon Book: Visual Symbols for Computer Systems and Documentation*. Wiley, New York.
- HOWE, J. AND O'SHEA, T. 1979. Computational metaphors for children. In *Human and Artificial Intelligence*, F. Klix, ed. North-Holland, Amsterdam. 141–153.
- HSU, Y.-C. 2006. The effects of metaphors on novice and expert learners' performance and mental-model development. *Interact. Comput.* 18, 770–792.
- IMAZ, M. AND BENYON, D. 2007. *Designing with Blends: Conceptual Foundations of Human-Computer Interaction and Software Engineering*. MIT Press, Cambridge, MA.
- IRBY, C., BERGINSTEINSSON, L., MORAN, T., NEWMAN W., AND TESLER, L. 1977. *A Methodology for User Interface Design*. Systems Development Division, Xerox Corporation.
- JOHNSON, G. J. 1994. Of metaphor and the difficulty of computer discourse. *Commun. ACM* 37, 12, 97–102.
- JOHNSON, J., ROBERTS, T. L., VERPLANK, W., SMITH, D. C., IRBY, C. H., BEARD, M., AND MACKAY, K. 1989. The Xerox Star: A retrospective. *IEEE Comput.* 22, 9, 11–26.
- JOHNSON, J. A. 1985. The desktop metaphor as an approach to user interface design (panel discussion). In *Proceedings of the CHI Conference*, 548.
- JOHNSON, M. 1987. *The Body in the Mind: The Bodily Basis of Meaning, Imagination and Reason*. University of Chicago Press, Chicago.
- JOHNSON, S. 1997. *Interface Culture: How New Technology Transforms the Way We Create and Communicate*. HarperEdge, New York.
- KAY, A. 1969. The reactive engine. Unpublished PhD Thesis, University of Utah. <http://www.mprove.de/diplom/gui/kay69.html>.
- KAY, A. 1972. A personal computer for children of all ages. Xerox PARC Research Rep.
- KAY, A. 1990. User interface: A personal view. In *The Art of Human-Computer Interface Design*, B. Laurel, ed. Addison Wesley, Reading, MA. 191–207.
- KAY, A. 1996. The early history of Smalltalk. In *History of Programming Languages II*, T. J. Bergin, Jr. and R. G. Gibson, Jr., eds. ACM, New York. 511–598.
- KAY, A. 2002. Graphical user interfaces. In *Software Pioneers: Contributions to Software Engineering*, M. Broy and E. Denert, eds. Lecture 5, DVD 2. Springer Verlag, Berlin.
- KAY, A. AND GOLDBERG, A. 1977. Personal dynamic media. *Computer* 10, 3, 31–41.
- KEESING, R. M. 1987. Models, “folk” and “cultural”: Paradigms regained?. In *Cultural Models in Language and Thought*, D. Holland and N. Quinn, eds. Cambridge University Press, Cambridge, UK. 369–393.
- KEMPTON, W. 1987. Two theories of home heat control. In *Cultural Models in Language and Thought*, D. Holland and N. Quinn, eds. Cambridge University Press, Cambridge, UK. 222–242.

- KOSARA, R., MIKSCH, S., SHAHAR, Y., AND JOHNSON, P. 2002. AsbruView: Capturing complex, time-oriented plans—Beyond flow charts. In *Diagrammatic Representation and Reasoning*, M. Anderson et al., eds. Springer Verlag, London. 535–549.
- LAKOFF, G. 1993. The contemporary theory of metaphor. In *Metaphor and Thought*, 2nd ed., A. Ortony, ed. Cambridge University Press, Cambridge, UK. 202–251.
- LAKOFF, G. AND JOHNSON, M. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.
- LAKOFF, G. AND JOHNSON, M. 1999. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Basic Books/Perseus.
- LAMPSON, B. 1972. Why Alto? *Xerox Inter-Office Memorandum*. <http://research.microsoft.com/~lampson/38a-WhyAlto/Acrobat.pdf>.
- LATOUR, B. 1987. *Science in Action*. Open University Press, Bristol, PA.
- LEACH, J. 2005. Disciplinary specialization and collaborative endeavour: Some challenges presented by sci-art projects. *Creativity and Practice Research Articles*, University of Dundee.
- LEARY, D. E. 1990. *Metaphors in the History of Psychology*. Cambridge University Press, Cambridge, UK.
- LEFRANCOIS, G. R. 1975. *Psychology for Teaching: A Bear Always/Usually Faces the Front*. Wadsworth, Belmont, CA.
- LEVY, S. 1994. *Insanely Great: The Life and Times of Macintosh, the Computer that Changed Everything*. Penguin, London.
- LICKLIDER, J. C. R. 1960. Man-Computer Symbiosis. *IRE Trans. Hum. Factors Electron.*, 1, 4–11.
- LICKLIDER, J. C. R. 1968. The computer as a communication device. *Sci. Technol.*, Apr.
- LIDDLE, D. 1996. Design of the conceptual model. Interview transcript in *Bringing Design to Software*, T. Winograd, ed. ACM, New York. 17–31.
- LINDSAY, R. K. 1988. Images and inference. *Cognition* 29, 3, 229–250.
- MCGRENERE, J., AND HO, W. 2002. Affordances: Clarifying and evolving a concept. In *Proceedings of the Graphics Interface 2000 Conference*, 179–186.
- MACKENZIE, D. 2001. *Mechanizing Proof: Computing, Risk and Trust*. MIT Press, Cambridge, MA.
- MCKIM, R. H. 1980. *Experiences in Visual Thinking*, PWS Publishing, Boston, MA.
- MADSEN, K. H. 1994. A guide to metaphorical design. *Commun. ACM* 37, 12, 57–62.
- MANDEL, T. 1997. *The Elements of User Interface Design*. Wiley, New York.
- MARCUS, A. 1994. Managing metaphors for advanced user interfaces. In *Proceedings of the AVI Conference*, 12–18.
- MARKOFF, J. 2005. *What the Dormouse Said: How the 60s Counterculture Shaped the Personal Computer Industry*. Viking Adult.
- MAYER, R. E. 1975. Different problem-solving competencies established in learning computer programming with and without meaningful models. *J. Educ. Psychol.* 67, 6, 725–734.
- MAYER, R. E. 1979. A psychology of learning BASIC. *Commun. ACM* 22, 11, 589–593.
- MAYER, R. E. AND GREENO, J. G. 1972. Structural differences between learning outcomes produced by different instructional methods. *J. Educ. Psychol.* 63, 165–173.
- MICROSOFT CORPORATION. 1995. *The Windows Interface Guidelines for Software Design*. Author, Redmond, WA.
- MOHNKERN, K. 1997. Beyond the interface metaphor. *SIGCHI Bull.* 29, 2, 11–15.
- MORGAN, C., WILLIAMS, G. AND LEMMONS, P. 1983. An interview with Wayne Rosing, Bruce Daniels and Larry Tesler. *Byte* 8, 2, 90–114.
- MOUNTFORD, S. J. 1990. Tools and techniques for creative design. In *The Art of Human-Computer Interface Design*. B. Laurel, ed. Addison Wesley, Reading, MA. 17–30.
- MÜLLER-PROVE, M. 2002. Vision and reality of hypertext and graphical user interfaces. University of Hamburg, Department of Informatics Rep. FBI-HH-B-237/02. <http://www.mprove.de/diplom/>.
- MYERS, B. A. 1998. A brief history of human-computer interaction technology. *Interactions*, Mar./Apr., 44–54.
- NELSON, T. H. 1990. The right way to think about software design. In *The Art of Human-Computer Interface Design*. B. Laurel, ed. Addison Wesley, Reading, MA. 235–243.
- NEWMAN, W. M. AND SPROULL, R. F. 1979. *Principles of Interactive Computer Graphics*, 2nd ed. McGraw-Hill, New York.

- NIETZSCHE, F. 1873/1999. On truth and lying in a non-moral sense. In *The Birth of Tragedy and Other Writings*, R. Geuss and R. Speirs, eds. Cambridge University Press. 139–153.
- NOBLE, J., BIDDLE, R. AND TEMPERO, E. 2001. Metaphor and metonymy in object-oriented design patterns. In *Proceedings of the 25th Australasian Computer Science Conference*, 187–195.
- NONOGAKI, H. AND UEDA, H. 1991. FRIEND21 project: A construction of 21st century human interface. In *Proceedings of the CHI Conference*, 407–414.
- NORMAN, D. A. 1988. *The Psychology of Everyday Things*. Basic Books, New York.
- NORMAN, D. A. 1999. Affordance, conventions, and design. *Interactions* 6, 3, 38–43.
- ORTONY, A., ED. 1993. *Metaphor and Thought*, 2nd ed. Cambridge University Press, New York.
- PAPERT, S. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Harvester Press, Brighton, UK.
- PERKINS, R., KELLER, D. S. AND LUDOLPH, F. 1997. Inventing the Lisa user interface. *Interactions*, Jan./Feb., 40–53.
- PIRHONEN, A. 2005. To simulate or stimulate? In search of the power of metaphor in design. In *Future Interaction Design*, A. Pirhonen et al., eds. Springer Verlag, London. 105–123.
- RACHOVIDES, D. AND SWIDERSKI, Z. 2001. *Workshop on Integrating Multimedia, Metaphors and Multimodality*. Proceedings of workshop on December 8th at PC-HCI. Patras, Greece. <http://www.comp.lancs.ac.uk/computing/research/mcg/mmm/program.php> (accessed 15 August 2005)
- RASKIN, J. 1994. Holes in history. *Interactions* 1, 3, 11–16.
- RASMUSSEN, J. 1986. *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. North Holland, New York.
- RHEINGOLD, H. 2000. *Tools for Thought: The History and Future of Mind-Expanding Technology*, 2nd ed. MIT Press, Cambridge, MA.
- RICHARDSON, A. 1993. The death of the designer. *Des. Issues* 9, 2, 34–43.
- ROBERTSON, G., MAARTEN VAN DANTZICH, M., ROBBINS, D., CZERWINSKI, M., HINCKLEY, K., RISDEN, K., THIEL, D. AND GOROKHOVSKY, V. 2000. The Task Gallery: A 3D windows manager. In *Proceedings of the CHI Conference*, 494–501.
- ROHRER, T. 1995. Metaphors we compute by: Bringing magic into interface design. Online report of University of Oregon Philosophy Department, Center for the Cognitive Science of Metaphor, <http://philosophy.uoregon.edu/metaphor/gui4web.htm>.
- RORTY, R. 1989. *Contingency, Irony and Solidarity*. Cambridge University Press, New York.
- ROTMAN, B. 1999. Thinking dia-grams: Mathematics and writing. In *The Science Studies Reader*, M. Biagioli, ed. Routledge, New York. 430–441.
- SCHILD, W., POWER, L. R. AND KARNAUGH, M. 1980. Pictureworld: A concept for future office systems. IBM Research Rep. RC 8384.
- SCHÖN, D. A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York.
- SCHÖN, D. A. 1993. Generative metaphor: A perspective on problem-setting in social policy. In *Metaphor and Thought*, 2nd ed. A. Ortony, ed. Cambridge University Press, New York. 137–163.
- SHASHA, D. AND LAZERE, C. 1995. *Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists*. Copernicus Press, New York.
- SHNEIDERMAN, B. 1983. Direct manipulation: A step beyond programming languages. *IEEE Computer*, Aug., 57–69.
- SMITH, B. C. 1996. *On the Origin of Objects*. MIT Press, Cambridge, MA.
- SMITH, D. K. AND ALEXANDER, R. C. 1988. *Fumbling the Future: How Xerox Invented, Then Ignored, the First Personal Computer*. William Morrow, New York.
- SMITH, D. C. 1977. *PYGMALION: A Computer Program to Model and Stimulate Creative Thinking*. Birkhäuser, Basel, Switzerland.
- SMITH, D. C. AND IRBY, C. H. 1998. Xerox Star live demonstration. In *Proceedings of the CHI Conference*, 17.
- SMITH, D. C., IRBY, C., KIMBALL, R. AND HARSLEM, E. 1982. The Star user interface: An overview. In *Proceedings of the National Computer Conference, AFIPS vol. 51*, 515–528.
- SMITH, D. C., IRBY, C., KIMBALL, R., VERPLANK, B. AND HARSLEM, E. 1982. Designing the Star user interface. *Byte* 7, 4, 242–282.

- SMYTH, M., ANDERSON, B. AND ALTY, J. L. 1995. Metaphor reflections and a tool for thought. In *People and Computers X, Proceedings of HCI'95*, M. A. R. Kirby et al., eds. Cambridge University Press, New York.
- STAR, S. L. 1989. *Regions of the Mind: Brain Research and the Quest for Scientific Certainty*. Stanford University Press, Stanford, CA.
- STENNING, K. AND OBERLANDER, J. 1995. A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Sci.* 19, 1, 97–140.
- STRAUSS, A. AND CORBIN, J. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage, Newbury Park, CA.
- SUTHERLAND, I. E. 1963/2003. Sketchpad, a man-machine graphical communication system. PhD Thesis at Massachusetts Institute of Technology, online version and editors' introduction by A.F. Blackwell and K. Rodden. Tech. Rep. 574. Cambridge University Computer Laboratory.
- SVANAES, D. AND VERPLANK, W. 2000. In search of metaphors for tangible user interfaces. In *Proceedings of the Conference on Designing Augmented Reality Environments*, 121–129.
- SWIGART, R. 1990. A writer's desktop. In *The Art of Human-Computer Interface Design*. B. Laurel, ed. Addison Wesley, Reading, MA. 135–141.
- TESLER, L. 2004. A CHI retrospective: 1962–1982. Talk given to SigCHI Puget Sound Chapter on June 24. <http://www.nomodes.com/TeslerPuget2004Jun24V.htm> (accessed 15/3/2005).
- TORENVLIET, G. 2003. We can't afford it. *Interactions*, Jul./Aug., 12–17.
- TREGLOWN, M. 1999. The role of metaphor in user interface design. Unpublished PhD Thesis, The Open University.
- TREGLOWN, M. 2004. HCI 2003 workshop reports: Metaphor and HCI. *Interfaces* 58 (Spring) 6–7.
- TURKLE, S. 1999. What are we thinking about when we are thinking about computers? In *The Science Studies Reader*, M. Biagioli, ed. Routledge, New York. 543–552.
- TURKLE, S. AND PAPERT, S. 1992. Epistemological pluralism and the reevaluation of the concrete. *J. Math. Behav.* 11, 1, 3–33. <http://www.articlet.org/articles/EpistemologicalPluralism.html>.
- VINCENTI, W. G. 1990. *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*. Johns Hopkins University Press, Baltimore, MD.
- VISSER, W. 2004. Dynamic aspects of design cognition: Elements for a cognitive model of design. Report 5144, INRIA, Rocquencourt.
- VAN DYKE, C. 1988. Binary jargon: The metaphoric language of computing. *SIGCSE Bull.* 20, 3, 34–41.
- WADLOW, T. 1981. The Xerox Alto computer. *BYTE Mag.* 6, 9, 58–68.
- WARE, W. 1996. Computing. In *Rand Corporation, Project AIR FORCE 50th Anniversary, 1946–1996*, 33–38. <http://www.rand.org/publications/PAFbook.pdf>.
- WEINSCHENK, S., JAMAR, P., AND YEO, S. C. 1997. *GUI Design Essentials*. Wiley, New York.
- WEISER, M. 1994. The world is not a desktop. *Interactions*, Jan., 7–8.
- WHITTAKER, S., TERVEEN, L., AND NARDI, B. A. 2000. A reference task agenda for HCI. In *Human-Computer Interaction in the New Millennium*, J. M. Carroll, ed. (2002). Addison-Wesley, New York. 167–190.
- WILLIAMS, G. 1983. The Lisa computer system. *Byte* 8, 2, 33–50.
- WINNER, L. 1986. Building the better mousetrap. In *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. L. Winner, University of Chicago Press, Chicago, 61–84.
- WOZNY, L. A. 1989. The application of metaphor, analogy and conceptual models in computer systems. *Interact. Comput.* 1, 3, 273–283.
- WYNNE, B. 1994. Public understanding of science. In *Handbook of Science and Technology Studies*. S. Jasanoff et al., eds. Sage, Thousand Oaks, CA. 361–388.
- YOUNG, R. M. 1976. Modelling cognitive development. In *Progress in Cognitive Development*. J. A. M. Howe and R. Young, eds. Department of Artificial Intelligence Research Report 17, University of Edinburgh.
- YOUNG, R. M. 1981. The machine inside the machine: Users' models of pocket calculators. *Int. J. Man-Mach. Studies* 15, 1, 51–85.
- YOUSEF, M. K. 2001. Assessment of metaphor efficacy in user interfaces for the elderly: A tentative model for enhancing accessibility. In *Proceedings of the EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly*. ACM, 120–124.

Received August 2005; revised February 2006; accepted March 2006