Little Machines: Understanding Users Understanding Interfaces

Johndan Johnson-Eilola
Technical Communications
Clarkson University
Potsdam, N.Y. 13699-5762
johndan@clarkson.edu

Abstract

This paper questions the ubiquitous practice of supplying minimalist information to users, of making that information functional only, of assuming that the Shannon-Weaver communication model should govern online systems, and of ignoring the social implications of such a stance. Help systems that provide fast, temporary solutions without providing any background information lead to the danger of users completing tasks that they do not understand at all. (Word will help us write a legal pleading, even if we have no idea what one is.) As a result, we have help systems that attempt to be invisible and to provide tool instruction but not conceptual instruction. Such a system presents itself as a neutral tool, but it is actually an incomplete environment, denying both the complexity and alternative (and possibly improved) modes of thinking about the subject at hand.

H.5.2 User Interfaces—*training, help, documentation*
Keywords: *online help, documentation, user interface*

> The space of a building is constructed to enclose something that must never appear within it. —Mark Wigley, *The Architecture of Deconsturction*

> An *ars oblivianis*? Forget it! —Umberto Eco

User friendly. User testing. Power user. In recognizing ourselves as computer users, we are also positioned (at least partially) as the used, the variable piece of the machine that closes the circuit, like a key in the ignition of a car. We are happiest when our technologies work automatically, when the machine anticipates our every desire. The machine is never completely absent from our attention, but it is becoming increasingly difficult—pointless, it seems—to think critically about the operation of the machine and our position within it. We don't think often about the ways in which technologies (and the larger, social technical system) construct positions that users assume, in effect, structuring users in specific ways. If the designers of programs have done their work well, users reason, then users shouldn't have to think.

Functional texts are defined by this politics of amnesia

Not that amnesia's a bad thing. Amnesia has become an operational requirement for our era. For information so inundates the lives of most users that they would literally be frozen if they could not routinely put aside important pieces of information. The inability to sort out, filter, and coordinate information—that is, to decide quickly what to forget—is one clinical definition of schizophrenia. Here, though, I want to return to the space covered over by routine acts of amnesia, to ask how users are constructed by computer texts (interfaces, documentation) as well as posit some implications of those constructions. Finally, I'd like to ask what other approaches avoid forgetting as a primary operational tactic: how can we help them actively build memories and experiences.

It's hard to argue with something that's not there

We pay a lot of attention to flashy technologies—multimedia presentations, real-time videoconferencing and document markup, vast hypermedia webs of global information—but our cleverest machines are invisible, used without thought, adapted and made part of our lives.

Figure 1: Balloon Help in Aldus Freehand

As Donna Haraway (1985) analyzes the situation, at their epitome, machines become intangible—"made of sunshine; they are all light and clean because they are nothing but signals...ether, quintessence" (p. 70). So while new technologies retain the capability to surprise us, eventually, successful technologies become commonplace, made invisible by their very successes.

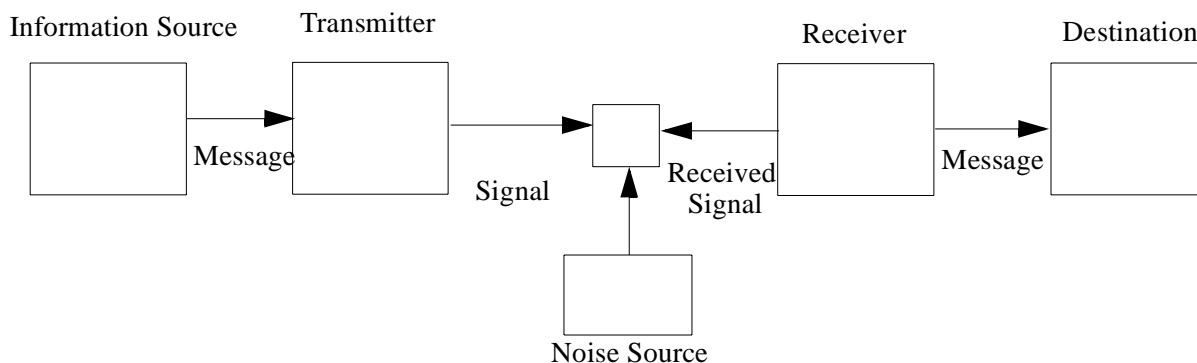*Figure 2: Tool Tip Help ("Style") in Microsoft Office Application*

Figure 3: Shannon and Weaver Type Communication Model

Print documentation seems doomed from its very outset—when a user focuses on a computer screen, any reference to a separate piece of media represents a failure of some sort: the computer wasn't transparent enough. There's truth to the saying, "If all else fails, read the manual." The printed document is a last resort.

Online documentation overcomes some of this issue by putting the manual within the technology. This act paradoxically makes manuals easier to access and more forgettable once we do use them. The limited successes of online documentation rely at least partially on the way that they call little attention to themselves. Where reading functional instructions in print requires retrieving books from shelves, locating and consulting information before returning to the "real" workspace, the interface, online documents at least collapse the media so that the help and the work are both within the interface.

Accepting Invisibility Restricts the Possibilities of Communication

Good online help, as it's typically defined, calls no attention to itself, asks the user to do very little. Although there are obvious reasons for this situation, we should not overlook some of the other implications. Among other things, when both designers and users accept invisibility as a goal in these systems (when, for example, users recognize themselves as the unspoken "You" of the command "Press the enter key") we participate in what's popularly known as the Shannon and Weaver model of communication from the 1940s, also sometimes called the transmission or conduit model: Information passes down a channel from sender to receiver. The receiver's job is this: *Present yourself as a target.*

Shannon and Weaver's model purports to offer a neutral, objective way of talking about communication. But, as the figure suggests, the model relies on a particular worldview, a scientific and mechanical version of communication and meaning. Not surprisingly, many people (in almost every field) have developed much more complex, socially situated models of the communication process that take into account the reader's role in the construction of meaning, the contingency of meaning, the context in which communication takes place, politics, and other factors. Shannon and Weaver in fact later complicated their own model by introducing channels for feedback; more recent approaches

have in turn provided more dynamic interaction loops—but the overall approach is still remarkably the same.

Why, then, are users still able to position themselves so easily in this simplistic model when they use online documentation? Why do designers of both interfaces and online help so insistently support this model? Simply because it works. Although Einsteinium physics replaced Newton's laws, people still apply Newtonian physics in their everyday experiences in the world and they get along just fine, thank you. So what if the old model is slightly off? It works well enough for most purposes. The key phrase is "works well enough": by defining the success of the project in terms such as speed and accuracy, such texts map out other concerns, from broader conceptual knowledge to the politics of technology.

Figure 4: Circa 1987 HyperCard Help Stack

So whereas early forms of online help attempted to naturalize themselves by appearing as books on screen, complete with spiral binders, index tabs, and a three-D look, designers and users have quickly discovered that what hypertext offers was not a way to improve on an old, slow-moving technology, but a way to create a new machine, one users occupy in order to navigate information space. (It's so fast it doesn't move.) Users are told by this machine that the Shannon and Weaver model works after all, once they have attained a level of technology powerful enough to support the (mainly one-way) process of communication. In print, the medium was the message, but that was always the problem with print—it got in the way. Online, we can make the medium disappear and leave the pure message (or so the argument goes).

The emphasis here on transparency in technical communication is not a surprising or even recent development. Technical communication has long been framed by its practitioners as an activity and discipline in which the medium should (ideally) be transparent:

Robert Connors' (1982) history of technical communication identifies the splitting off of technical communication from English departments as due in part to the heightened sense of a need for efficiency in functional and technical prose (p. 332). And David Dobrin (1983), while maintaining a critical stance on both the fluidity and power of definitions, notes that "technical writing's greatest success comes when it is swallowed easily and digested quickly" (p. 247).

Issue 1: Do as I say, not as I do

Easy to use computer systems present designers and users with a paradox: Users don't want to struggle to master arcane command names and codes. At the same time, most communicators know that communication is a complex, recursive set of processes involving writers, readers, and their contexts. So we insist that the Shannon and Weaver model is somewhat outdated, but we reaffirm it constantly. It works, because we accept a narrow vision of how to measure the success of these online texts. This should be our first clue: We, as a field, tend to understand the use of documentation as a recursive, active, creative process: users don't simply receive information; they skim it, paraphrase it, connect it up to their previous experiences, experiment with it, remake it. But we also encourage our users to think of documentation, online or print, as approaching invisibility. So even as our field increases the complex-

ity of communication as a practice, the vision we construct for our audiences contradicts that complexity, makes our work and us invisible.

Issue 2: Real learning disappears in the collapse of time

Computer documentation performs both training and teaching functions. That is, to some extent users will always refer to documentation as a system that assists them in simply operating a technology as if it were a hand tool. To make a simple, straightforward analogy, when someone picks up a hammer, they don't expect to need to know the finer details of cabintry if all they need to do is drive a nail.

But is this always true? Shouldn't using a hammer, in a well-designed system, help users eventually learn more complex carpentry skills? In order for a user to learn those advanced skills, they need to apprentice themselves to another, more experienced carpenter or avail themselves of more complex educational situations—workshops, trade schools, etc. The majority of users never move from training to learning.

To bring the analogy back around, most documentation supports training, but not learning. In the case of the hammer, there were significant technical issues that prevent the hammer from supporting learning (obvious issues like the fact that there's no Help button on a hammer, not flexible display, etc.). Computers, though,
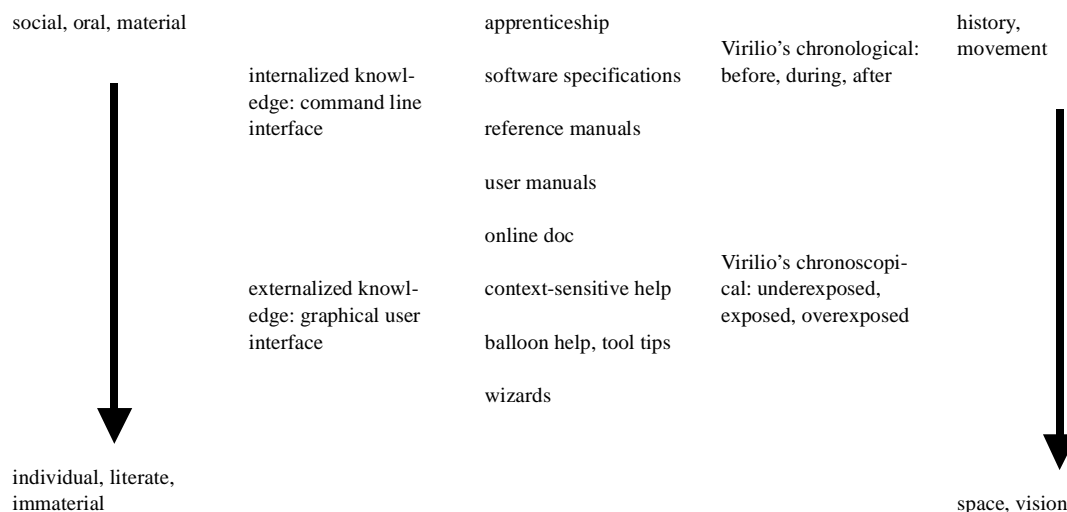
Figure 5: Rough History of Help Systems

social, oral, material — apprenticeship — history, movement

internalized knowledge: command line interface — software specifications — Virilio's chronological: before, during, after

reference manuals

user manuals

online doc

externalized knowledge: graphical user interface — context-sensitive help — Virilio's chronoscopical: underexposed, exposed, overexposed

balloon help, tool tips

wizards

individual, literate, immaterial — space, vision

can clearly support the transition from training to learning.

Indeed, many of us have worked on tutorial and self-paced learning systems that teach broad skills in communication, management, design, and more.

These popular exceptions aside, though, documentation of mass-market systems steadfastly refuses to move from training to learning.

A second difficulty with functional documentation-and interface design as a whole-is the tendency to collapse critical distance in the pursuit of increased efficiency. Documentation, however, is frequently valued precisely because it can seem to act instantaneously. Learning, however, requires more time than training.

Roughly speaking, we can track an evolutionary movement in documentation away from three characteristics (social, oral, and physical) and toward three opposing poles (individual, literate, invisible).

Wizards, near the bottom of the chart, constitute an interesting potential counter-category: potentially more complex responses to help queries. Wizards might be used to help users learn rather than simply locate information as quickly as possible. However, as I'll discuss below, wizards are most often used to artificially reduce the complexity of a user's, to automate things that probably aren't very amenable to automation.

We can see the ways in which assistance moves from outside of the machine toward the inside, and from outside specific applications and into them. Help is now presented to users as a part of the workspace itself. Not only has online help conquered the tedium of walking to a bookshelf and manually finding pages, but now context-sensitive forms of help and iconic cues about possible actions act to remove even the act of navigation from using online help-it's just there when you need it. This is hard to argue with: If I had a choice between a two-word, on-screen prompt about the function of a tool and the alternate task of finding a print manual, locating the relevant information through the use of a table of contents or index, and then navigating to the information (and reading it), I'd probably try the five-word description (and I'm more willing to waste time than the average user).

Perhaps more alarming, though, are the cases where the machine does attempt to function in contexts where simple adjustments and binary choices will not do. Style-analysis programs are one example. Newer forms of help attempt to automate teaching and learning to

Figure 6: Sample Screens from Microsoft Word Resume Wizard

the point that the activity disappears. In Microsoft's interestingly named "wizards," for example, users construct documents based on a series of basic questions and standard templates. Word ships with standard wizards for memos, press releases, resumes, agendas, and even one for a legal pleading letter.

The resume wizard, for example, walks users through the layout of standard resumes, letting them select among numerous resume classes (am I "professional" or "contemporary" or "elegant"?) and stock and optional categories, as well as custom section headings.

There are certainly benefits to this arrangement, but I'm concerned about the fact that wizards don't attempt to offer much in the way of advice about why one would choose some headings over others, for example.

And the only response I can articulate toward the resume wizard is that, like style-analysis programs, it may provide the context for a useful discussion about why computer programs fail at some tasks. However, most users will not be prompted to engage in those discussions (those who teach computer documentation or writing might help students engage in this discussion, but the majority of Microsoft Word users are not in classes; many of those who are in more academic settings may be in areas that hold simplistic models of writing).

Another Wizard in Word walks users through formatting a Legal Pleading document (but does not discuss what it is or how to write it). I'm all for people learning the types of knowledge that are too frequently held only by the elite, but I don't see where this Wizard helps the type of learning that actually lets a person write an effective legal pleading. It seems more than a little legally dangerous to begin writing these things without background knowledge and, furthermore, no attempt by the system to help the user gain that background knowledge. (In the overactive theater screen in my mind, I see a cartoon version of Joe Pesci playing the lead role in Microsoft's deceased social agent in the legal drama/comedy, *My Cousin Bob*.) The same hods true for the other wizards.

Issue 3: At the speed of light, time ceases to be an issue

A somewhat more complex problem appears as the result of the computer's emphasis on increased speed and the collapse of critical distance, something urban planner and social theorist Paul Virilio described as a reaching an "absolute speed" in which everything that's important is immediately accessible. In such systems, Virilio argues, we seem to pilot a sort of "static audio-visual vehicle" that gives us "the condition of possibility of a sudden mobilization of the illusion of the world, of an entire world, that is telepresent at every moment." This is the paradox of speed, where objects moving at the speed of light no longer experience time.

In one sense, computer users do navigate from place to place, moving from the desktop, from folder to folder, across disks, into word-processing, graphics, video, and audio programs, and even out to the network, where they traverse the World Wide Web and enter into other user's computers. But, "navigation" puts perhaps the wrong spin on what I see happening here: users are not going anywhere. Rather, the world is brought to them. As Virilio ironically puts it,

[W]ith the revolution of instantaneous transmissions, we are witnessing the beginnings of a type of general arrival in which everything arrives so quickly that departure becomes unnecessary.

This situation leads to a couple of potentially troubling (but probably familiar) problems, such as the impulse for computer programs to move menu items (which are accessible but relatively invisible) into the interface itself in the form of toolboxes (as with Aldus FreeHand, Figure 1 above) and button bars (as with Microsoft Word, Figure 2 above). Symptomatically, these movements foster the idea that everything important is visible, that the distance between desire and result should be near zero, that the World Wide Web really extends inclusively over the entire world. Certainly, these are only tendencies, and ones that are commonly reversed by other factors, but there appears to be a clear shift toward what we might call interface inclusiveness.

Re-articulation: Socialize online help

Documentation used to be non-existent: users were enculturated into a community of users by experts. Unfortunately, as many of us have found, dealing with the experts has not always been easy: they're frequently possessive, speak discourses other than our own, and not interested in the same things we are. In addition, there's not frequently enough experts to go around. Historically, documentation (like all print) rehearsed the movement from human master/apprentice relations to private consultation with a text. Rather than asking

someone to teach you how to do something, you use a text.

Still, using documentation is still typically the last resort-users are more likely to ask each other for help rather than consult a text (online or print). I have a hard time recalling an instance in which any of my students consulted a printed document unless I forced them to. Ironically, or perhaps tellingly, these same students are either professional writing students or computer science students, many of whom will be employed writing such documentation.

The difficulty of most online help is that it explicitly isolates users from each other. I'm not calling for a return to the traditional master/apprentice model, because its structure skews power away from learners and toward masters. But, a collaborative model of online help might allow users to work with each other, contributing advice or asking questions based on their own varying levels and areas of expertise. The strength of the little machines model makes this idea seem a little odd—who, after all, would want to spend their time answering questions about software, design, complex troubleshooting configuration, or writing processes?

But if we look at the willingness with which people do engage in such discussions in existing online forums—WebBoards, instant messaging channels, USENET lists, MOO/MUD spaces, chat rooms, email discussion groups—we can begin to see the possibility of poking holes in the barriers that construct online help as an isolated, individual space.

Re-articulation: Get users to understand interfaces as interested maps

Cartographer Denis Wood argued that the power of a map was its "ability to do work." Maps are not simply neutral, passive representations—they are active, they channel, structure, and document actions within the world. A map of the world documents the results of civil wars and diplomatic agreements about boundaries; it also structures activities in the world, directing participants to go to one location but not another, to enforce one set of laws in one region and a different set in another. So on one hand, yes, maps are only representations of the real world (I cannot write the name "Johndanland" in place of "Canada" and thereby own Canada), but to the degree that they are sanctioned and believed by communities, they structure how participants in those communities act (by participating within

the legal system of property ownership in the state of New York—including property taxes—my wife and I can have our name written on a specific, five-acre parcel of land located in the township of Hopkinton). The necessary connections between the social and the physical networks involved in these mappings give maps more power than we typically allow them. (But anyone who has initiated a title search as part of a property purchase understands the importance of such maps, both graphical and textual.)

Woods writes that pieces of computer documentation work like maps; they "operate effectively... [T]hey don't fail. On the contrary they succeed, they achieve effects, they get things done." At the same time, both maps and documentation "make present—they represent—the accumulated thought and labor of the past... [M]aps facilitate the reproduction of the culture that brings them into being."

Working from, among other things, Wood's theories about the politics of cartography, Cindy and Dickie Selfe (1994) have argued that contemporary interface mappings rely heavily on Eurocentric, corporate ways of seeing and working. Folders, clocks, limited alphabets, hierarchical filing systems all work to validate one particular worldview. By conceiving of interfaces as (inter)texts, Selfe and Selfe say, we might begin helping ourselves and users recognize the interested and political nature of interfaces, and also begin working to construct other representations.

Similarly, we might begin questioning the assumptions that allow online texts to operate mechanically: What exactly is being automated? What decisions are made, and who/what is making them? If this task wasn't automated, what would I have to know to make the choices myself? What other texts and people might this text be connected to? There are other possible articulations of "machine":

> A book itself is a little machine; what is the relation (also measurable) of this literary machine to a war machine, love machine, revolutionary machine, etc.—an abstract machine that sweeps them along? We have been criticized for over-quoting literary authors. But when one writes, the only question is which other machine the literary machine can be plugged into, must be plugged into in order to work.... Literature is an assemblage.
> Deleuze & Guattari, "Rhizome" (p. 4)

Re-articulation: Combine functional instruction with conceptual instruction

As I alluded earlier, one way in which documentation short-circuits questions and critical reflection is the strict division typically maintained between "tool" instruction and "conceptual" instruction: Online documentation normally offers instructions for tools (how to indent a paragraph, how to change a font from one family to another), but ignores conceptual instruction. A potential reason for this ignorance is the fact that such pedagogical discussions would have to admit that writing was not easy, a position that flies in the face of the image being portrayed. Early online documentation stressed brevity—not merely in sentence structure but in the sheer amount of material provided—because of disk space restrictions. But, given the demands of contemporary versions of Microsoft Office, it's hard to believe disk space is still an issue.

In many of our own classes, we take up this discussion in the form of broad learning—discussion of layout, the processes of writing, etc. And it could be argued that the machine will never be able to replace a human teacher. But documentation in general is founded on the idea that one cannot always have a teacher physically present. We write textbooks, design handouts, pair students up with one another for peer critiques. Conceptually sound texts on issues like Website design increasingly cover issues of process, audience, etc. (see, for example, Rosenfeld and Morville's *Information Architecture of the World Wide Web* or Johnson-Eilola, *Designing Effective Web Sites*). We should recognize that even when computer-supported advice is not without its own problems (style-analysis programs, for example), we should also work to help reconstruct these supports because they will be there whether we like it or not.

Re-articulation: We might also begin thinking about the crucial difference between "automation" and "selective focus"

Automation makes user intervention difficult or impossible. Industrial anthropologist Larry Hirshhorn, for example, recounts the experiences of a group of machinists who taught themselves how to program an automated lathe and, as a result, began improving the parts made by the machines (rather than, as the machinists had previously done, merely loading material into the machine and removing parts after they were made).

Management, however, viewed the workers as operating in areas outside of their qualifications, and inserted a mechanism on the machine to prevent the machinists from reprogramming it.

Similar structures are often present in documentation, typically because of the argument that conceptual information or elaborations are "extraneous." The resume wizard I showed earlier, for example, offers a minimal number of choices to users but in no way encourages them to expand on those choices, or rework them in new ways. The wizard does offer limited advice on when to use chronological rather than skills based organizations, but it does not offer any discussion beyond that. But why limit it to five words? Why not connect those fragmented, telegraphic bits of information to extended information? Why not, in fact, connect the five-word descriptions to full-blown lessons on page design or writing processes, or whatever? Because the priorities of online help-compact efficiently functioning-prevent such important things as offering the user any choice that cannot be addressed without thought. Online help addressing complicated design issues such as the appropriate use of hanging indents, kerning, leading, etc. must necessarily (according to the efficiency model) only offer functional instructions. The technology disguises itself as a neutral tool rather than an incomplete environment, never suggesting that the user might want to think about the operation or learn background theories. The implication behind online help in most computer programs is that the user already knows the theory behind the work, and that the computer is only a neutral tool (often complicated but, ideally, obvious). In this situation, the "problem" of writing or other complex activities is "solved" not by dealing the complexity of that situation, but denying that it exists.

But as technical communicators, if we consider our work to be helping users improve the quality of their own lives and cultures, we have to do more than cover over or deny the fact that life is complicated. Instead, we must help users understand communication, production, thinking, and living as an often messy, complicated, open-ended activity, one that often requires attention to not merely the simplest functional activities but also the larger frameworks and contexts of that work.

A very early version of this article was presented at the 1995 Midwest Conference of the Association for Business Communication.

References

Connors, Robert J. (1982). The rise of technical writing instruction in America. *Journal of Technical Writing and Communication* 12, 329-352.

Deleuze, Gilles, and Félix Guattari. (1987). *A thousand plateaus: Capitalism and schizophrenia*, (Brian Massumi, Trans.). Minneapolis: U of Minnesota P.

Dobrin, David N. (1983). What's technical about technical writing? In Paul V. Anderson, R. John Brockman, & Carolyn R. Miller (Eds.), *New essays in technical and scientific communication: Vol. 2. Research, theory, practice* (pp. 227-250). Farmingdale, NY: Baywood.

Eco, U. (1988). *An Ars Oblivionalis? Forget it!* PMLA 103, 254-261.

Haraway, Donna. (1985, March/April). A manifesto for cyborgs: Science, technology, and socialist feminism in the 1980s. *Socialist Review* 80, 65-105.

Hirschhorn, Larry. (1984). *Beyond mechanization: Work and technology in a postindustrial age*. Cambridge: MIT Press.

Johnson-Eilola, J. (2001). *Designing effective websites*. Boston, MA: Houghton-Mifflin.

Rosenfeld, L. and P. Morville. (1998). *Information architecture of the World Wide Web*. Sebastopol, CA: O'Reilly.

Selfe, Cynthia L. & Richard J. Selfe. (1994). The politics of the interface: Power and its exercise in electronic contact zones. *College Composition and Communication* 45, 480-504.

Shannon, Claude E. & Warren Weaver. (1949). *The mathematical theory of communication*. Urbana: U of Illinois P.

Virilio, Paul. (1986) *Speed and politics: An essay on dromology* (Mark Polizzotti, Trans.). New York: Semiotext(e).

Wigley, M. (1995). *The architecture of deconstruction: Derrida's haunt*. Cambridge, MA: MIT Press.

Wood, Denis. (1992). *The power of maps*. New York: Guilford.

Woolever, Kristin R. & Helen M. Loeb. (1994). *Writing for the computer industry*. Englewood Cliffs, NJ: Prentice-Hall.