

Conversations on Context: Some Incomplete Thoughts

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

Abstract

Johnson-Eilola responds to commentaries by noting that all ask, in different ways, the same question he is asking: How far can and should we go in supplying greater context and complexity to software users?

H.5.2 User Interfaces—training, help, documentation

Keywords: online help, documentation, user interface

I want to begin by thanking the three respondents for their careful and valuable readings of "Little Machines." They've provided some important and interesting ways for rethinking, extending, and challenging our field's approaches to computer documentation and to communication and living in general. My primary goal in drafting the initial article (the first draft of which was in 1995) was to begin to open up this issue for discussion: when should interfaces teach and when should they simply train?

This question would seem to be a trivial one. And perhaps it is: it would not represent the first trivial issue that's stumped me. But my concern, developing over the past fifteen or more years, has centered on the fact that computer documentation, and software in general, tend to automatically prioritize training and automation over teaching. And, as I ceded in "Little Machines," there are certainly innumerable instances in which automation is desirable. But just as importantly, there are also many, often extremely important, contexts in which the amnesia requisite within automation can be limiting or even damaging to users. In general, we have done a good job (and continue to improve at) analyzing and implementing solutions that help users complete their "real" work. We have done less about figuring out what to do when users lack complex knowledge that would seem to be important to the success of their "real" work. Too often, our response to complex and/or varied user contexts has been to decontextualize the user's activity until, like a sharp stone rolled by a river's current, the complicating factors are worn away, leaving us with a smoothly polished stone. How do we judge the quality of a stone? It's impossible to say, without context.

I can offer, as should be evident by now, no simple and easy solution to this dilemma. As Kathy Haramundanis rightly points out, documentation specialists frequently do not possess the time or the domain knowledge necessary to deal fully with these complex situations. But, she continues, "time con-

straints [are] not a good argument for omitting basic conceptual information."

There's an inherent complexity here, an unsolvable problem, one that each of the repentantly both agrees with and deals within in different ways. Although I find both points of agreement and disagreement with each, I am not going to undertake a point by point response, because to do so would place us on the terrain of debate rather than conversation and dialogue. I am not interested in Truths with capital T's at this point, but only in opening up this discussion. With that goal in mind, I'd like to provide some quick, generalized responses that I hope will help JCD readers become engaged in the discussion as well.

To some extent, the whole discussion revolves around the issue of the location of responsibility for understanding and dealing with the complexity of contexts within which software and documentation are used. Haramundanis and Selber locate that responsibility primarily in technical communicators (both as individuals and as a discipline); Farkas locates the responsibility primarily within users.

As Haramundanis points out, users are often not equipped to deal with the complexities of situations they work within. This is nothing new: the experience of life is, almost by definition, one of challenge and growth. We all learn to cope with new experiences and, more importantly, learn from those experiences. But, as she continues, what do we do when software gives users a false sense of mastery? Don't technical communicators have an ethical responsibility to warn users to the potential problems? As she suggests, at the very least we need something akin to warning labels to help users understand that things may be more complex than they think. In some cases, references to specific external documents and even to human teachers may be required.

As she warns us, even the rudimentary task of recognizing when users might be in invisibly complex situations requires a great deal of skill. Although, movements such as interaction design (along with contextual inquiry and participatory design) offer some suggestions about new ways of sharing responsibility for learning.

Selber, similarly, argues that educational institutions are not doing an adequate job of preparing technical communicators to think about complex social issues. On one hand, he recognizes that traditional approaches to usability provide important feedback to many aspects of designing computer-based texts (his primary

example revolves around Websites); on the other hand, traditional approaches often look primarily at technical features, a move "by which the whole complex of usability comes to be represented exclusively by certain instrumental aspects of it". This is a key observation: the argument here is not that attention to instrumental aspects is mistaken but that social (and other) aspects should not be ignored. Selber's curricular materials represent an important example of ways that thinking about technology use (of which computer documentation is one typical component) can be expanded in socially important ways. The specifics of Selber's prototypical student, while not immediately translatable to every institution, suggests ways to open up education about computer documentation. By helping students see the social aspects of their work rather than simply the instrumental ones, they can begin the complex and long-term task of generating new approaches, new sensibilities, and new media that deal more effectively with users' different situations and needs.

If Haramundanis and Selber locate the responsibility for understanding and representing complexity within the field of technical communication, Farkas flips the issue on its head. Noting my own assertion that users read actively, he questions how passive and accepting users really are: "users of documentation regularly form judgements (often harsh ones) about the documentation and the company that made the software, and they regularly depart from the guidance offered by the documentation." To which I absolutely agree: users are smarter and more demanding than we frequently assume they are.

I am grateful to Farkas for having raised this issue, since it's one that many of us frequently slip into. Users are not dupes, led around mindlessly by interfaces and documentation. They skim, they reread, they provide localized examples and test them, they purchase third party manuals (a market that's spawned a whole industry).

At the same time, though, we need to recognize that the overwhelming majority of users' activities with a particular piece of software rely on "default" settings and approaches. For every person who purchases a third-party manual with robust explanations and detailed examples, an order of magnitude more use only the print manual or online help. And for every user who routinely uses the Help menu to access online tutorials or help systems, several thousand make guesses based on interface elements such as tooltips, button names, menu labels and dialogue boxes (and,

not understanding how to evaluate their successes or failures except in decontextualized ways, assume they're correct). So the situation is much more complex than I probably initially portrayed it, but the tendency is still, I believe, accurate: users want to understand complexity, but they are culturally and technologically discouraged from doing so. The problem is not merely that they don't have time to learn more complex but more powerful ways of working and communicating: it's that they're not offered methods for doing so, or even a rationale.

When the entire computer industry seems directed at making users think that software solves real-world problems with elfin magic, is it surprising that in most cases users accept decontextualized representations of their work? Resistance is possible, with much effort and in relatively isolated instances, but in general the status quo prevails. That is the job of the status quo, as social theorists like to point out: the commonality of "common sense" makes it damned near impossible to question, let alone change. I'm not suggesting that we automatically complicate every situation beyond comprehension, only that we begin to question some of these situations and work to invent new responses and structures.

We are beginning to see some important (if still hesitant) gestures in this at complicating these issues, which Farkas reminds us of toward the end of his response. More recent editions of Microsoft Office template materials, for example, refer to texts for additional learning about resumes. And users themselves increasingly resort to the Internet for locating additional learning materials and communicating with other users, including experts, to help them more fully understand the complexities of their work. But we need to be

sure that these sorts of complications, explanations, and elaborations are available (although not forced up) to users in ways much more heavily integrated into their experiences. In the end, the effort to help contextualize computer use, to enmesh it within its varied and multiple social contexts, remains a key point to which I am committed; I am gratified to see that goal played out in both complex and important ways by each of the three respondents¹.

The computer industry's generalized response to complex user contexts has been, by and large, the attempt to decontextualize them. Teachers, theorists, and practitioners in computer documentation have, by and large, gone along with this practice (often out of the necessity of survival). The rhetorically complex act of writing, for example, is worn away until it becomes the mere act of rearranging letters and fonts on screen, a small round stone to hold in the palm of your hand. One might ask, as Farkas, Selber, and Haramundanis do, how far we can go in helping users (and even ourselves) understand the complexity of innumerable contexts and complex knowledge domains. That's a good question, but one to which I don't have a short answer. But it's an issue I think our field has an ethical responsibility to explore, both for and with users.

¹ I want to note here that my earliest public work in this area (a panel presentation at ACM SIGDOC in 1992 in Ottawa that included Stuart Selber and Dickie Selfe, among others) was met with almost universal disdain by the audience, who voiced many of the complex concerns voiced by Selber, Haramundanis, and Farkas here; but rather than recognizing the importance of these goals, the earlier audience thought the problems insurmountable and the goals misguided.