

1

Introduction: Tyrants, Heroes, and Victims in Information Design

At a police station in the Midwest, a police officer named Barbara starts up the DOS-based database that she will use for locating and analyzing traffic accidents in a particular area. According to the software's manual, she should first unroll a three-by-three-foot map of the area, which is overlaid with six-digit numerical coordinates called node numbers. Then she should look up the node number for each intersection she is investigating and type them, one by one, into a dialog box. The cumbersome map is rich in unnecessary detail, takes the entire space of a cleared desk, and must be held down by paperweights so that it will not roll back up; it's no surprise that Barbara avoids using it. Instead, she opens a folder and takes out a Post-It™ note on which she had written down a series of node numbers some months before. The unwieldy node map is replaced by a conveniently sized note that holds only the details that she needs.

A common trope in the literature of user-centered design is the worker-as-victim: the everyday Joe or Jane who is oppressed by an unjust tyranny and in need of rescue. The tyrannical system could be imposed by selfish, shortsighted employers (Bravo 1993¹), an antidemocratic capitalist system (Ehn 1989; Bjerknes and Bratteteig 1995), managers who do not understand or are opposed to the needs of workers (Bødker 1991; Gronbæk et al. 1993), a flawed work structure (Coble et al. 1996; Holtzblatt and Beyer 1996; Ramey, Rowberg, and Robinson 1996), poorly designed tools that do not take into account how workers actually get things done (Dumas and Redish 1993; Gronbæk, Kyng, and Mogensen 1993; Rubin 1994), or even a theoretical stance (Johnson

1998). The worker-as-victim is portrayed as needing to be rescued by a heroic figure, an information designer. This heroic figure is enlightened, principled, and capable, and is able to employ user-centered design methods to defeat the tyrannical system and rescue the victims—sometimes through the invention of a benevolent work structure (Beyer and Holtzblatt 1998), sometimes by providing a better tool for accomplishing work activity (Bødker and Gronbæk 1996), sometimes by emancipating workers through organizing labor in a class struggle with management (Ehn 1989, 1993; Bjerknes and Bratteteig 1995; see Spinuzzi 2002e for a review), and sometimes by providing a more sensible theoretical system (Johnson 1998). But in any case, the designer listens to the worker-victims, synthesizes their comments and feedback, and develops the means of their rescue. The resulting solutions, it is asserted, lead to sustainability (Hackos, Hammar, and Elser 1997; Hackos and Redish 1998), worker empowerment (Beyer and Holtzblatt 1998; Bravo 1993; Clement and Van den Besselaar 1993; Ehn 1993), and the examination of users and technology use from their perspective (Johnson 1998). Designers strive for a well-considered system that, if properly constructed, will liberate workers who desperately need to be rescued. Indeed, Geoff Cooper and John Bowers (1995) note that human-computer interaction research in general, and user-centered design in particular, often justifies and legitimizes itself through this sort of “compassionate discourse.” “It is important to recognize the rhetorical functionality of these characterizations of the user for HCI,” they tell us. They add that “it is not so much that users are angry, frightened, and different from designers, it is more that, for this way of legitimizing HCI, they *have to be*” (p. 51).

But in the quotation that opened this chapter, Barbara is not waiting around to be rescued. Although the software is not set up to facilitate the particular tasks in which she is engaged, she does not wring her hands and wait for an information designer to come slay the dragon. She picks up available tools, adapts them in idiosyncratic ways, and makes do. Through these “invisible” innovations (Nardi and Engeström 1999; Suchman 1995) she subverts the information system, inventing her own ways to turn it to her needs.

Workers like Barbara tend to create their own practices, tools, and texts constantly, sometimes in cooperation with the existing information

systems, sometimes in competition with them. For instance, Mark Zachry (2001a; see also Spinuzzi and Zachry 2000) describes computer users who co-opted managerial genres such as administrative memos to customize computer documentation. Christian Heath and Paul Luff (2000) relate how medical practitioners, construction workers, and personnel working for the London Underground rejected the use of computer technology for keeping records, relying instead on older paper documents that were more mobile and accessible. And Barbara Mirel (1988) notes that workers in her study avoided relying on official documentation by developing unofficial ways of sharing information, such as via intraoffice cliques and residential experts. As Geoffrey Bowker and Susan Leigh Star (1999, 159; see also Star 1995) conclude, “Imposed standards will produce workarounds. Because imposed standards cannot account for every local contingency, users will tailor the standardized forms, information systems, schedules, and so forth to meet their needs.”

The messiness of everyday work life—the unofficial, unpredictable ways workers assert their own agency, turn to their own problem-solving skills, and individually or cooperatively design practices, tools, and texts to deal with recurrent problems—is reflected in a considerable number of thoughtful studies (e.g., Kyng and Mathiassen 1997; Nardi 1996). But as Button and Dourish (1996) point out, the problem comes about when attempting to link naturalistic studies, which describe these local innovations, with design methods, which translate the findings into design work. As I argue later in this chapter, many of the most popular user-centered design methods assume that the goal of research is to inform *centralized solutions*; they assume that design solutions must spring from, or at least be ratified and promoted by, decision makers with specialized knowledge. Even when unofficial user innovations have been proven useful, researchers working within these approaches tend to take such innovations as rough solutions to common underlying problems, solutions that should be officially refined and consolidated by a trained designer if these underlying problems are to be truly solved. If individuals such as Barbara have developed an innovative way to get work done, these designers might examine that unofficial innovation primarily so that they can develop an official, approved, standardized version that everyone can use. The operating assumption is that if innovations are to

be effective, the worker-victims' many coping strategies must be united and refined by the designer-hero.

This operating assumption is particularly troubling to me because it is gaining a foothold in my own discipline, rhetoric, particularly in the subfield of technical communication. User-centered design approaches have correctly been seen as promising new avenues for analyzing and understanding audiences, but as I argue later in this chapter, the user-centered design approaches that have most often been adopted are those that cast workers as victims and designers as heroes.

In this book I propose a new understanding of technologically mediated work, for information designers in general and particularly for rhetoricians and technical communicators. I turn away from the trope of the worker-as-victim and its tendency to minimize or officialize workers' innovations. Instead, I place these innovations at center stage: I examine the crucial subversive interactions in which workers routinely engage as they use information systems to accomplish their activities. I do this not to heroically rescue the workers from a patronizing and disempowering trope; as we will see, they often do a pretty good job of "rescuing" themselves. Instead my goal is to better understand why information design so often fails to catch on and become sustainable, why workers so often alter the designed artifacts (particularly textual artifacts) they are presented with, and how designers might approach design tasks as true partnerships that result in designs flexible enough to be adopted.

To pursue these issues, I outline an alternative field methodology for investigating designed artifacts in the context of work activity—*genre tracing*—and illustrate it through four studies of diverse workers in a loose network of governmental agencies. Genre tracing bears a strong resemblance to the activity theory-based approaches that have been gaining ground in human-computer interaction and computer-supported cooperative work (e.g., Nardi 1996). However, it is particularly suited to applications in rhetoric and technical communication because it draws from rhetorical theory and makes texts (in the broad sense) central to its investigations. Although I do not go the extra step of outlining a design methodology in this book, in the final chapter I discuss some implications that this book has for information design.

At the same time, I believe that trained information designers can contribute much to the emergent innovations of workers, not by replac-

ing those innovations with centralized solutions, but by helping to design systems that workers can modify. This book should not be read as advocating quietism, the notion that systems can and should always repair themselves. In fact, workers' innovations are disparate and often of the chewing-gum-and-bailing-wire variety; without designers to periodically consolidate and rethink these solutions, any given activity can begin to disintegrate as disparate solutions lead groups of workers in different directions. Rather, this book offers a methodology that can ideally encourage trained information designers and innovative workers to enter into true partnerships.

In this opening chapter, I first describe the user-centered design discussion as it is making its way into technical communication. I am especially interested in how the victimhood trope is used to justify the adoption of user-centered design approaches in technical communication. I then examine how a subset of user-centered design methods, what I call *fieldwork-to-formalization* methods, positions users in its characterization of workers, fieldwork, and formalizations. Next, I draw on M. M. Bakhtin's discussion of official and unofficial discourse to frame an alternative methodology: genre tracing. Finally, I briefly describe the genre tracing methodology and how it positions users.

“WRITERS, WRITERS EVERYWHERE”: POSITIONING THE USER IN TECHNICAL COMMUNICATION

Technical communication is based in rhetoric, but it also draws from psychology, anthropology, sociology, and related approaches. Lately, technical communicators have also sought to align their field more closely with information design (e.g., Geisler et al. 2001; Hart-Davidson 2001) and other interdisciplinary fields such as human-computer interaction and computer-supported cooperative work. This realignment has led technical communicators to adopt methods from the related disciplines. And user-centered design methods appear to be a strong match, since they combine a humanistic mission of advocating for the audience, new empirical approaches to the ancient art of audience analysis, and strong frameworks for translating audience insights into design suggestions. These themes come together, for instance, in Smart and Whiting's contextual design study of office software. “Frequently,” they note,

“technical communicators view themselves as users’ advocates, with the mistaken notion that, as nondevelopers who also use an application designed by someone else, they know what users want and need.” But Smart and Whiting’s team found that “if they truly wanted to become user advocates, they needed direct contact with users” (2002, 159), contact that they translated into information design through a popular user-centered design approach.

The introduction of user-centered design to technical communication is instructive because of the ways it is positioned and justified. In introducing user-centered design to technical communication, scholars have sometimes characterized information design as dichotomized between *user-centered design* and its opposite, *system-centered design*. Although I discuss these two approaches in more detail below, it is important to keep two things in mind about how these approaches are characterized. First, they are characterized as totalizing: every design approach and every evaluation of designed information can be categorized as being on one side or the other of the system-centered/user-centered divide. Second, few if any technical communication scholars advocate a system-centered view; system-centered design functions almost exclusively as a straw person, a demonstrably poor choice in contrast with the more favored user-centered design. (See Mirel 1998a for a related argument.)

In technical communication, user-centered design has been examined most extensively by Robert Johnson in his book *User-Centered Technology* (1998; for other technical communication literature on user-centered design, see Dumas and Redish 1993; Hackos and Redish 1998; Rubin 1994; Schriver 1997; Wixon and Ramey 1996). In the discussion that follows, I use Johnson’s book to explore the methodological assumptions that underlie user-centered design as it has been represented in technical communication.

Methodological Assumptions of User-Centered Design

As Patricia Sullivan and James Porter (1997, 11) point out, in research literature the terms *method* and *methodology* are often used interchangeably. Although these terms deal substantially with the same subject—the question of how to bring a coherent approach to research—they express

quite different things. A *method* is a way of investigating phenomena; a *methodology* is the theory, philosophy, heuristics, aims, and values that underlie, motivate, and guide the method. The distinction is important to keep in mind as we explore the methodological assumptions underpinning user-centered design as it has been represented in technical communication. As we will see, these methodological assumptions include the trope of worker-as-victim, and that trope shapes the methods and the sorts of things one might expect to learn from them. Furthermore, user-centered design is implicitly portrayed as the sole alternative to system-centered design.

Johnson sets up the dichotomy between system-centered design and user-centered design quite clearly, arguing that “the user-centered view is philosophically and practically at the opposite end of the spectrum from the system-centered view” (p. 129; see also p. 30). (Johnson uses system-centered design as a straw person for providing a contrast with user-centered design; in fact, he admits that there are few advocates of system-centered design (p. 124).) He draws several comparisons between system-centered and user-centered design throughout the book, both explicitly and implicitly (see especially pp. 25–33). Three of these are listed below:

- Whereas system-centered design is formalist (p. 25), user-centered design is social constructionist (p. 93).
- Whereas system-centered design is rationalist, determinist, and modernist (pp. 25–27), user-centered design is postrationalist, nondeterminist, and postmodernist.
- Whereas system-centered design involves centrally controlled design (pp. 25–27), user-centered design involves collective, cooperative design (pp. 30–32).

Indeed, the methodological assumptions of the two design approaches appear to be just about as far apart as they can get—binary opposites. Perhaps the most important comparison is the first one. In Johnson’s view, system-centered design is founded on formalist thought, “based upon models of technology that focus on the artifact or system as primary, and on the notion that the inventors or developers of the technology know best its design, dissemination, and intended use.” It perceives technology, people, and context “as constituting one system

that operates in a rational manner toward the achievement of pre-determined goals” (p. 25). Johnson asserts that user-centered design, on the other hand, is founded on social constructionist thought, which is “based on the concept that reality is mutable, that there are no certain truths, and that knowledge is constructed through communally created knowledge and action.” In this view, technology “can be interpreted and reinterpreted depending on the people involved, the context or situation in which it is designed, developed, or deployed, and the historical moment it resides within” (p. 93).

Johnson’s concern is with how people are empowered or disempowered by the design of texts and other technological artifacts. He wants to “examine users and the phenomena of technological use from their perspective” (p. 4). In short, Johnson sees users as sociopolitically empowered through the help of a designer (in this case, a writer), someone who identifies with the users and who has the authority and skill to transform the way they perform their goal-directed actions—authority and skill that the users themselves do not have. Indeed, the things that Johnson promises to do in the book include:

- Recognizing and understanding how “cogs” of society actually have valuable, detailed knowledge (p. 61)
- Revealing moments of human knowledge and the essence of human involvement with technology (p. 132)
- Determining which medium will best fit the user situation and tasks (p. 133)
- Providing audience analysis for underpinning design work (p. 145)

Johnson’s goals are laudable, reflecting the humanist values that underpin technical communication and opposing the disempowering effects of system-centered, Taylorist approaches. Yet in the way they are framed, these goals assume the victimhood trope. They position users as unable or unqualified to undertake design work on their own. Users are not cast as agents who initiate and implement change themselves. Despite the assertion, often repeated in user-centered design circles, that the user is a codesigner (e.g., Beyer and Holtzblatt 1998; Salvo 2001; Wixon and Ramey 1996), *users do not actually control the design*, either in Johnson’s book or in the wider technical communication literature on

user-centered design. At most, they nominate ideas that the designer then might choose to ratify during the final design of the artifact. Although Johnson criticizes system-centered design for assuming that designers of technology “know best its design, dissemination, and intended use” (p. 25), his brand of user-centered design makes the same assumption. The difference is that whereas system-centered designers rely solely on their own knowledge of the system, user-centered designers also draw on their compassionate studies of the users.

For instance, in their book on contextual design—the user-centered design method that has gained the most solid foothold in technical communication (see, e.g., Beabes and Flanders 1995; Hackos, Hammar, and Elser 1997; Hackos et al. 1995; Raven and Flanders 1996; Smart and Whiting 2002; Smart, Whiting, and DeTienne 2002; Wixon and Ramey 1996)—Beyer and Holtzblatt (1998, 370) say that they want to “co-design the system with the users.” But they make clear that this co-designer status is relegated to describing work and providing feedback. Users should emphatically not be expected to understand the designers’ work models, which describe the users’ activities (p. 369); “It’s their job to do their job, not design systems” (p. 371). And their innovations and feedback are useful only when designers consolidate and shape them to support the work models that the users are not able to understand. While users give valuable ideas during the prototyping phase, for instance, the designer “is free to think up a better mechanism” (p. 400).

As we have seen, Johnson casts workers as needing rescue. His list of goals calls for designer/writers to understand, identify with, and analyze the users so they can bequeath empowerment to the worker-victims more effectively; reveal the knowledge that has remained hidden even from the users themselves; determine the optimal media for users’ tasks; and analyze users as an audience, the more-or-less passive receivers of the designer/writer’s information. The workers are positioned as victims unable to rescue themselves.

Methods

Johnson turns to the methods of participatory design to accomplish his list of goals (see pp. 82–83). Again, these goals are worthwhile and

compelling (as is Johnson's book itself), but they are based in methodological assumptions that position the user as a victim to be rescued.

For instance, Johnson advocates that writers empower users by localizing and redesigning one sort of tool—documentation—in such a way as to codify and formally document the word-of-mouth knowledge and practices in a company (p. 149). The writers—a great number of “writers, writers everywhere,” in fact—would thus rescue users by consolidating informal, unarticulated practices into official, formal, and authoritative documents. This process would have to be undertaken carefully, by trained² writers, since the traditionally inflexible genres of computer documentation structure and constrain users' work in undesirable and disempowering ways (pp. 140–141).

An illustration is provided by Johnson's description of a study undertaken by his students, in which they observed and interviewed an individual secretary as she walked through various tasks with a software manual. The study involved the writers examining the user's work, then utilizing the results to help redesign the manual. But it did not involve inviting the user to codesign the manual or examining how the user supplemented the documentation with other practices and artifacts.

To sum up, as technical communicators have adapted user-centered design approaches to their own field, they have drawn heavily on the aspects that emphasize the victimhood trope. The trope is a natural match to technical communication's focus on humanistic ethics, but it ultimately leads to a paternalistic relationship between designers and workers. This relationship is particularly reflected in fieldwork-to-formalization methods such as the user-centered design method most popular with technical communicators, contextual design (Beyer and Holtzblatt 1998).

FIELDWORK-TO-FORMALIZATION METHODS: OBSERVING WORKERS, MODELING BEHAVIOR

In much user-centered design work—particularly in a variety of coherent methods developed in the United States and Britain to help organizations rapidly design information—descriptive, naturalistic studies of actual work have been paired with abstract work models. This pairing re-

flects the interdisciplinary nature of user-centered design. The descriptive studies, patterned after the ethnographies used in anthropology and sociology, are meant to unearth the workarounds, innovations, and tacit practices workers have developed. The models, often drawn from management or systems design approaches, are suitable for generalizing, standardizing, regularizing, idealizing, and managing work, as well as for providing brief descriptions to systems designers. It is an uneasy pairing, one that assumes that researchers can easily move from the particular to the general, from divergent local practices to a single ideal model of the work. This disconnect is particularly strong in methods that, in this book, I will call *fieldwork-to-formalization methods*. Examples of fieldwork-to-formalization methods include contextual design (Beyer and Holtzblatt 1998); the research stage of joint application design (JAD; see Wood and Silver 1995); client-led design (Stowell and West 1994); and user-centered information design (Henry 1998); and to a lesser extent coherence (Viller and Somerville 2000) and other applications of rapid ethnography (e.g., Millen 2000). They vary in detail on both the fieldwork and modeling ends. For instance, Millen's approach pairs rapid ethnographies with informal, innovated causal models; JAD turns informal interviews and observations into highly formalized models; and contextual design has a high degree of detail on both ends.

Fieldwork-to-formalization methods are "meant to guide system design through the stages of gathering data from customers, modeling and interpreting that information, and designing and implementing systems based on that information" (Bisantz and Ockerman 2002, 263). That is, they bridge field studies (including naturalistic work observations, unstructured interviews, and analysis of artifacts used in the work) and information design through models or through categorical and sequential descriptions of the work. In doing so, they span boundaries between *organizations* (organizations that need information systems and organizations that produce them; see Korpela, Mursu, and Soriyan 2002) and between *disciplines* (workplace researchers and information designers; see McCarthy 2000). Furthermore, since these methods involve working within short engineering cycles, the data gathering is typically compressed and the analysis is done primarily through the same models used to communicate the results to systems developers (Macaulay, Benyon,

and Crerar 2000; for examples, see Beyer and Holtzblatt 1998; Wood and Silver 1995).

Many researchers have called into question the assumption that moving from fieldwork to formalization is unproblematic. For instance, in an issue of *Communications of the ACM* devoted in part to workplace investigations, Liam Bannon (1995, 66; see also Sachs 1995) objects that formalizations are too reductive to capture the nuances of the fieldwork:

The argument is not whether some level of abstraction and formalization of work processes is possible or desirable, but rather, whether such techniques could in principle capture all that is required, and how to manage what is inevitably left outside the representation. While some simply argue for more powerful representational forms, there has been a growing awareness that the problem is not simply one of richer notations or more ample resources but, more fundamentally, of an inappropriate concept of what can, in principle, be captured in any model of the work process.

So what *does* get captured in these models of the work process? As Yrjö Engeström (1999b, 63, 64) notes in his critique of business process reengineering (BPR)—a movement that is aligned with fieldwork-to-formalization methods (Wood and Silver 1995) and that shares their tendency to optimize work—“Attempts at making everyday practices of work visible are driven by different motives. In various management techniques, the overriding motive of visibilization is control.” That motive leads, in Engeström’s words, to “a quest for complete rationality and elimination of unnecessary steps.” Work-process models tend to represent just such a managerial or organizational view of the work, as Patricia Sachs (1995) points out in her own critique of BPR. This view emphasizes the overall workings of an organization in generalities suitable for regularizing and rationalizing work; it assumes that work has an underlying structure that, once described, can be made more efficient. That is, the goal of such models is not to value workers’ innovations themselves but to take them as symptoms of an underlying problem that can then be solved by manipulating the model—to rescue the workers from an inefficient system and empower them to meet management’s goals. Naturally, these models are rarely made accessible to the workers whose work they describe.

Below, I discuss in more detail how fieldwork-to-formalization methods position and portray workers; describe the conduct of field-

work; and translate the results of fieldwork into formalizations such as models, categories, and sequences.

Positioning the User: The Victimhood Trope in Fieldwork-to-Formalization Methods

Like many other user-centered design approaches, fieldwork-to-formalization methods often justify themselves through the victimhood trope. In these methods, designers offer workers freedom from their victimhood, but *victimhood* is conceived as coming from barriers to doing their jobs efficiently, and freedom consequently comes through a process in which their work is increasingly managed, regularized, and rationalized. Their workarounds and innovations are examined, formalized, modeled, collapsed with similar innovations, and finally mandated by the new system—or supplanted by other practices that the designer has determined are better. In other words, this sort of freedom comes through compliance with an increasingly formalized and rationalized work process in which workers may have input, but little or no final say. Workers enjoy *functional empowerment*, in which they are empowered to perform their tasks in a prescribed manner, rather than *democratic empowerment*, in which they have a decision-making role in how their organization operates and how technology fits into their jobs (see Clement 1994; Blomberg, Suchman, and Trigg 1997).

For instance, the contextual design literature emphasizes understanding and empathy for workers (synonymously called “users” or “customers”). “When we participate in the users’ world,” Holtzblatt and Beyer (1993, 94) say, “we want it shown to us so well that we know it—we want our feet to be sore where their shoes pinch.” In fact, contextual design texts frequently include stories and scenarios describing how designers should put themselves in the workers’ shoes. But these stories and scenarios, and their solutions, tend to focus on functional empowerment: how to improve the workers’ efficiency and productivity by redesigning artifacts and practices. For instance, in the introduction to *Contextual Design*, Beyer and Holtzblatt (1998, 6) describe “the true story of one user trying to do a simple task: A user of a standard office system needs to print a label.” After describing the user’s frustrating

efforts and her eventual abandonment of the task, Beyer and Holtzblatt conclude that “this system supports work poorly. It is poor not because functions are missing but because the system imposes a work model that does not make the job more efficient and does not match the user’s expectations” (p. 7). A properly designed system, they argue, “provides an optimal match between the users’ current way of working and the work practice introduced by the new system; it changes the work enough to make it more efficient but not so much that people cannot make the transition” (p. 8, their emphasis). The words *efficient* and *optimal* make frequent appearances throughout the rest of the text, underscoring contextual design’s commitment to functionally empowering the workers.

Other fieldwork-to-formalization methods similarly position workers as victims of inefficient systems. In his description of user-centered information design, Henry (1998) lists a variety of users’ reactions to “unusable” software, such as confusion, frustration, panic, and boredom. The most telling negative reaction is “*misuse or modification*. Those who know the software well may change it to meet personal requirements that do not advance organizational interests” (p. 7). That is, workers’ innovations are positioned as a dangerous symptom of workers’ victimhood. Similarly, Stowell and West (1994, 22, 29) describe client-led design as a way to avoid “a feeling of insecurity in those most affected,” which can be manifested in “lack of cooperation, refusal to use the new information system, sabotage, withdrawal of goodwill, and industrial action.” With client-led design, the underlying management problems are “unraveled” and clients buy into the process, which results—at least in the case study that Stowell and West present—in a more efficient company that produces new, high-quality products (Stowell and West 1994, chaps. 6–7).

The victimhood trope, then, is often used to underpin fieldwork-to-formalization methods. Specifically, these methods position workers as sharing management’s goals of efficiency and work intensification, but describe the workers’ attempts to “do their jobs” as being frustrated by poorly designed work processes and the information artifacts that support them. Workers’ innovations are portrayed at best as symptoms of the underlying problem, and sometimes, uncharitably, as wrongheaded and fumbling attempts to resist the system. These innovations are rarely

depicted as valuable solutions in their own right and never as solutions that can be allowed to remain under the control of their originators, never as locally grounded practices that can be adopted or rejected at the individual worker's discretion. Design problems are portrayed as systemic and systemwide, meaning that solutions should be implemented at the same scope. Consequently, fieldwork-to-formalization methods typically seek to rectify problems through a compassionate, efficiency-focused investigation of workers' actual work practices and artifacts, followed by a reductive modeling of the fieldwork in management's terms so that designers can develop the most optimal work structure and artifacts.

Which is not to say these workers are being victimized by fieldwork-to-formalization methods. I am not going to appeal to the same trope that I am critiquing! What I want to drive home here is that these methods attempt to fit workers into their story of designerly heroism, while at the same time workers like Barbara are quietly "rescuing" themselves by tailoring workarounds to their local situations.

Gathering Field Data

Given their goal of investigating actual work practices, fieldwork-to-formalization methods draw on a range of fieldwork techniques. These range from extremely informal to somewhat formal methods.

On the informal end of the scale, the research stage of JAD involves JAD facilitators visiting sites, talking with individual workers or groups of workers, looking at artifacts at the interviewees' work location, and informally observing work (Wood and Silver 1995, chap. 5). Facilitators are encouraged to ask workers about their business objectives (such as increasing productivity, decreasing costs, and improving customer satisfaction; see p. 54) and to look out for "distractions" and other breakdowns in work flow (p. 57); they are not encouraged to examine workarounds. Facilitators are not trained in fieldwork.

In the middle of the scale, contextual design involves a highly developed set of techniques that represent the adaptation of "ethnographic research methods to fit the time and resource constraints of engineering" (Holtzblatt and Beyer 1993, 93), including unstructured and semi-structured interviews, walkthroughs, and artifact analysis (Beyer and

Holtzblatt 1998, chaps. 2–4; Raven and Flanders 1996). Designers are advised to look for the work’s underlying structure and for opportunities to build designs on that existing work structure. Innovations are cast as symptoms of problems and as starting points for global redesign efforts. In this case, investigators are trained in three-day sessions. Similarly, client-led design involves action research, which—as Stowell and West (1994) use the term—mainly consists of interviews in which investigators focus on how various workers construe their organization’s problems.

On the more formal end of the scale, coherence and rapid ethnography both involve observational fieldwork by trained researchers, although the fieldwork takes far less time than in the case of standard ethnographies. As Viller and Somerville (2000, 171; see also Millen 2000) say about coherence, “The approach should not be construed as ‘ethnography-lite’, some cut down or simplified version of ethnography. Rather, the method is informed by cumulative experience of applying ethnographic approaches to the development of requirements for computer-based systems.”

These fieldwork approaches have come under attack by trained ethnographers, who complain that fieldwork is a difficult and problematic exercise demanding long training. For instance, Diana Forsythe (1999, 136; see also Cooper et al. 1995; Nyce and Lowgren 1995) critiques how “do-it-yourself ethnography” has been used by investigators untrained in ethnographic methods, including systems designers and content experts:

The problem is that in ethnography as in some other pursuits, a little knowledge can be a dangerous thing: superficial social research may confer the illusion of increased understanding when in fact no such understanding has been achieved. This problem is illustrated by the nature of recent do-it-yourself ethnography in medical informatics [specifically, a Contextual Design project], in which brief exercises in shadowing, observation, and interviewing have been undertaken from a common sense stance without engaging the questions that define ethnography as anthropologists understand it. Such an exercise can result in a cognitive hall of mirrors. Without addressing basic issues such as the problem of perspective, researchers have no way of knowing whether they have really understood anything of their informants’ world view or have simply projected and then “discovered” their own assumptions in the data.

As we saw in the previous subsection, these assumptions are likely to include (1) an underlying work structure that can be reified for the pur-

pose of redesigning work and (2) the presence of workarounds like Barbara's as symptoms of problems in this underlying work structure.

These fieldwork-to-formalization methods share an explicit focus on the actual practices of workers, but the focus tends to be in the service of the victimhood trope. It leads investigators to examine how workers' efficiency is compromised by work practices and artifacts, and views workarounds as symptoms of underlying problems or, at best, rough solutions for the designer to improve and standardize. The methods do not act to sustain, enable, or understand innovations as a vital part of the work.

Building Formalizations

After the fieldwork come the formalizations—the models, categorical descriptions, and sequential descriptions—that can be used to communicate findings to software developers and to describe and design future systems. The object is to zero in on the specific aspects of the work that will affect the redesign. (As Macaulay, Benyon, and Crerar (2000, 40) observe, “Ethnographers tend to report their findings in lengthy monographs. Systems designers are thought to like diagrams with as little text as possible, although our own experiences have not entirely supported this presumption.”) Formalizations are used to consolidate the field data and find overall patterns that might shed light on the underlying work structure. That is, they serve to rationalize work.

The formalizations used by the different methods tend to differ widely. On one end, client-led design (Stowell and West 1994) uses ten different high-detail formalizations culled from soft system methodology, structured system analysis and design method, object-oriented analysis, and other sources; on the other, user-centered information design (Henry 1998) employs task description, information-use models, and other relatively low-detail formalizations. Despite their differences, these formalizations tend to consolidate the innovations noted in the fieldwork and construct unified models of the underlying work structure. In doing so, they provide the designers with that which makes them heroes: a special viewpoint on the work that is inaccessible to the workers themselves, one that qualifies them to detect, ratify, and improve the best innovations of

the workers. This managerial viewpoint focuses on values such as efficiency and work intensification. Take for instance this passage, in which Holtzblatt and Beyer (1993, 97–98) describe how to fold workers' innovations into the redesigned system:

Our best ideas for improving the work often come from seeing how a particularly thoughtful person or group has solved their own problems. We build this solution into our abstract work models and our system, so all customers can take advantage of it. Once we have this consolidated model, we study it for problems and inefficiencies. We bring together data from all customers, keeping good ideas, fixing problems, and using technology to combine steps. When done, we have a statement of how our users will work, if we can implement the system to support it.

The consolidated model is not accessible to the workers themselves. As Beyer and Holtzblatt (1998, 369) say, “In Contextual Design, we don’t even try to talk to our customers with our work models” because doing so would involve training these workers in this “new language” that explicitly describes aspects of work that the workers have never been able to articulate on their own. The work of reading models and translating them into design cannot (and should not) be left to the workers, who do not (and should not) have the means to rescue themselves: “Customers aren’t technologists—they don’t know the range of possibilities that technology could support. They may be either unrealistic or excessively cautious as a result. And they don’t know what it takes to make a design hang together. And why should they, after all? It’s their job to do their job, not to design systems” (p. 371).

Since the workers are not qualified to rescue themselves, they must rely on the heroic design team, whose members are trained in reading these models and are capable and principled enough to turn them into humane design decisions. Workers are allowed to give input on the system, but the designers know best how to design “a coherent response that hangs together as a new work practice” (Beyer and Holtzblatt 1998, 305).

OFFICIAL AND UNOFFICIAL SOLUTIONS

I should make clear at this point that it is not a bad thing for information designers to study how people do their work and to design artifacts and practices that might facilitate that work. Information designers, includ-

ing technical communicators, *should* be trained, principled, and capable user advocates, and they *should* understand how workers are often constrained and disempowered by existing tools and ways of doing work.

Fieldwork-to-formalization methods have gone a long way toward these goals, and they offer plenty of success stories describing how workers are happy with the results of projects based on them. But these methods leave little room for examining worker agency. They pass over or try to control the unofficial, idiosyncratic, ad hoc solutions such as Barbara's innovation described at the beginning of this chapter. These methods are guided by the managerial goal of a *normative solution*: a tool or set of work practices that, once codified and optimized, can functionally empower the worker-victims.

The trope of worker-as-victim, I contend, devalues the multiple and innovative solutions that workers like Barbara develop, tends to paper over the contingencies to which workers continually adjust, and leads researchers to develop and use analyses that minimize the role of such contingencies while maximizing the role of commonalities in work. As I argue in chapter 2, fieldwork-to-formalization methods tend to assume some sort of structure that underlies the work of a range of workers, a structure that can be investigated, modeled, and repaired in such a way as to solve the workers' general problems. That structure might take the form of contextual design's work models (Beyer and Holtzblatt 1998), the tasks employed in usability testing (Dumas and Redish 1993), and so forth. In these cases, the data collection and analysis methods are designed to shift attention away from local exigencies and toward common problems and common solutions. Workers' innovations are seen as symptoms of an underlying problem; the researcher's role is to pin down that problem and the designer's role is to develop an idealized solution, a solution that may incorporate, but ultimately obviates, workers' local innovations. Thus they tend to minimize the agency of those workers—and miss some of the important differences in how workers undertake and conceptualize their work.

On the other hand, trained designers can avoid common pitfalls of workers' homegrown solutions, which tend to be of the chewing-gum-and-bailing-wire variety. Workers produce solutions that are devious, wily, and cunning, but often these solutions do not involve a deep

understanding of the system, and sometimes they even run to superstition. Workers produce solutions that work—but often they do not produce solutions that work well *by their own criteria*, and often those solutions are not promulgated so that other workers can take advantage of them. The drawbacks of these ad hoc solutions, of course, lead designers to attempt to formalize or officialize them. As Star (1995, 111) puts it,

Organizations attempt strategies that will try to create organizational consistency in the face of strong tensions between formal representations and empirical experience. The tensions arise from the fact that ad hoc strategies, work-arounds, and local knowledge that keeps organizations going [i.e., unofficial solutions] are first deleted from formal representations [i.e., official solutions]. When the formalizations become recipes for action, then further ad hoc work-arounds are necessary to make the prescriptions fit the local circumstances.”

Star warns that “this can be an infinitely recursive process” (p. 111). To examine this dynamic, ever-shifting balance between designers’ contributions and workers’ innovations, I turn to language philosopher M. M. Bakhtin and his distinction between the official and the unofficial (1981, 1986).

Bakhtin argues that two competing impulses shape how we communicate: the centripetal and the centrifugal. The *centripetal* impulse is toward formalization, normalization, regularity, convention, stability—and stasis. Things are metaphorically drawn to the center and become official. In contrast, the *centrifugal* impulse is that of resistance, idiosyncrasy, ad hoc innovation—and chaos. Things metaphorically fly away from the center and become unofficial (Bakhtin 1981, 270–273). As Morson and Emerson (1990, 30) put it, official forces “seek to impose order on an essentially heterogeneous and messy world” while unofficial forces “either purposefully or *for no particular reason* continually disrupt that order.”

Note that we are not dealing with a simple two-dimensional continuum or binary opposition. In this metaphor, centripetal force draws things in from all sides. Centrifugal force, on the other hand, pulls things outward in all directions: “Centrifugal forces are a panoply of the most heterogeneous elements. They may have no relation to each other except their divergence of the ‘official’” (Morson and Emerson 1990, 30).

In these terms, fieldwork-to-formalization methods tend to be centripetal: they tend to normalize behavior and tools to produce centrally controlled, official solutions. On the other hand, workers' innovations tend to be centrifugal in that they resist a centralized system inadequately adapted to their particular, situated needs. Such innovations start out as idiosyncratic and unofficial solutions, often involving unconventional genres or unconventional genre usage. For instance, Barbara's Post-It note (what Bakhtin (1981, 273) surely would call a "low genre") was an opportunistic use of a ready-to-hand artifact and a basic genre, the handwritten list.

Yet there is no sharp line between official and unofficial innovations (Morson and Emerson 1990, 30). Like designers, workers feel the centripetal impulse and officialize solutions. For instance, handwritten lists like Barbara's were used by other workers I observed, and we can imagine that over time such lists could become relatively standardized (though still handwritten) so that they could be more easily shared and interpreted by the community of workers. Without the centripetal impulse, unofficial solutions can lead to chaos: imagine a new worker who is asked to work with the private, idiosyncratic filing system of another worker. A system that has become too officialized can be inflexible and rule-bound, unable to adapt to change, and unwilling to grant agency to workers; a system that has become too unofficial can be too flexible and chaotic, resistant to conventional approaches, and deficient in organizational memory and coherence. Typically, though, organizations avoid these extremes (as they must, if they are to continue functioning) and maintain a dynamic tension between centripetal and centrifugal impulses.

Neither centripetal nor centrifugal impulses are inherently wrong-headed. Indeed, fieldwork-to-formalization methods have often yielded strong designs that work better than the systems they replace, as many cases attest. But these methods attempt to replace local, idiosyncratic, or contingent solutions with universal, standardized ones. In other words, officialization entails consolidating flexible, rapidly developing solutions with less flexible, slowly developing, more extensively codified ones. The result might be a system that is closed, static, unable to accommodate

local contingencies or changes because its components shut off productive linkages with unofficial innovations. The computerized information system described by Heath and Luff (2000), for instance, was so closed that workers could not find ways to link it with their own innovations, and they ended up abandoning it altogether. And even if workers find a closed system to be initially useful, their activities constantly change and diverge. A closed information system, like a perfectly tailored suit, “fits” only as long as its subject does not change.

Elsewhere Mark Zachry and I have discussed an open-systems design approach that attempts to balance official and unofficial solutions (Spinuzzi and Zachry 2000). In this book, I use the official-unofficial distinction to guide workplace research. I return to the question of design in the final chapter.

CONCLUSION

As I have suggested above—and as I will discuss in more detail in chapter 2—since fieldwork-to-formalization methods assume that workers are unable to empower themselves, the goal of research is to inform the development of idealized artifacts, work practices, and work structures meant to standardize work in ways that functionally empower workers. Certainly these methods sometimes entail examining user innovations, collecting feedback, and even collaborating with users to redesign artifacts. But in the end, the goal is to transform a messy set of ad hoc, unofficial solutions into a single, neat, coherent, official—and static—generalized solution.

If we are to study the dynamic tension of centripetal and centrifugal impulses rather than papering over the idiosyncrasies of users’ unofficial solutions, an appropriate research methodology is needed. I contend that this methodology should be based in sociocultural theory, yet be connected solidly to existing research methods. *Genre tracing* is one such approach. Based in activity theory and genre theory, genre tracing draws on established methods that have been used with those theories. Genre tracing provides a way to highlight users’ experiences with official and unofficial genres and to compare them across communities or workplaces.

Genre tracing is *dialogic* (Bakhtin 1981)—it draws on the metaphor of dialogue to examine how people interact with complex institutions, disciplines, and communities; how they solve problems and disseminate solutions; and how their conversations and problem solving are instantiated in artifacts. Genre tracing is concerned with examining the ways that workers rescue themselves—if that is indeed an appropriate metaphor—by developing unofficial, frequently unarticulated work practices and genres, by adapting old genres to new uses, and by linking their innovations to established, official genres.

Genre tracing draws on existing research methods, including many employed in user-centered design approaches, but repurposes them under different methodological assumptions. Genre tracing's methods are thus accessible to information designers. And since genre tracing draws on established methods, studies based on genre tracing can be held to similar standards of repeatability, reliability, and validity.

Genre tracing can be time consuming and labor intensive—just as ethnographic research, ethnomethodological research, and fieldwork-to-formalization methods can be. Like these other research approaches, genre tracing is best used at a major turning point, such as the beginning of a major design or redesign project (see chapter 2), and conceivably could be conducted in concert with these other research approaches. For instance, a genre tracing project could conceivably share the data collected in a contextual design project, although it would analyze those data in considerably different ways.

In the next chapter, I argue that the assumption of an underlying work structure, the methodological assumption that is so central to the attempts designers make to consolidate and officialize users' innovations, is deeply embedded in field-to-formalization methods. This assumption is problematic for multiple reasons. I then outline genre tracing as an alternative methodology for conducting workplace investigations.