HUMAN
COMMUNICATION RESEARCH

Human Communication Research ISSN 0360-3989

ORIGINAL ARTICLE

Why Do People Reject New Technologies and Stymie Organizational Changes of Which They Are in Favor?

Exploring Misalignments Between Social Interactions and Materiality

Paul M. Leonardi

Department of Communication Studies, Department of Industrial Engineering and Management Sciences,
Northwestern University, 2240 Campus Drive, Evanston, IL 60208

This article explores the relationship between users' interpretations of a new technology and failure of organizational change. I suggest that people form interpretations of a new technology not only based on their conversations with others, but also through their use of technology's material features directly. Through qualitative and quantitative analysis of ethnographic data on the implementation and use of a computer simulation technology at a major automotive firm, I show that engineers' communication with managers, coworkers, and customers led them to develop an interpretation about what the technology was supposed to do while their interactions with the material features of complementary technologies led them to develop an interpretation that the new simulation technology was not an efficient tool for that specific purpose. I show how the interpretations developed from people's material interactions moderate the effects of the interpretations developed through social interactions on willingness to use the technology in the future. I then demonstrate that, in this particular setting, engineers inadvertently stymied an organizational change of which they were very much in favor by reducing their use of the new technology. I conclude by discussing how misalignments between the information generated in users' interactions with others and with technologies' material features can lead to the failure of planned organizational change.

doi:10.1111/j.1468-2958.2009.01357.x

Most new technology implementations in organizations are accompanied by an agenda for workplace change. Many managers decide to employ new information technologies with the hope that the material features of those technologies will alter workers' existing communication patterns and thereby help them to communicate

---

Corresponding author: Paul M. Leonardi; e-mail: Leonardi@northwestern.edu

more effectively (DeSanctis & Monge, 1999; Leonardi & Bailey, 2008; Rice & Gattiker, 2001) or provide them with the time and means to communicate about new issues or topics (Leonardi, 2007; Majchrzak, Rice, Malhotra, King, & Ba, 2000; Markus, 2004). Indeed, a good deal of research shows that information technologies are often conceived of with specific organizational change agendas in mind (DeSanctis & Poole, 1994; Jackson, Poole, & Kuhn, 2002).

Unfortunately for many managers, implementing a particular technology—even one designed specifically to bring about change—is no guarantee that organizational change will occur. If people resist a technology that provides material features that must be used to enable new patterns of communication and interaction, they will often lack the capabilities to communicate differently and planned organizational change will likely fail. Thus, it seems that to explain why technology-driven organizational change does not occur, paying close attention to the reasons people resist a new technology, is extremely important for both theory and practice (Davidson, 2006; Rice, 1991; Robey & Boudreau, 1999).

Some researchers argue that people resist a new technology because they are dispositionally disposed to do so: They do not like change, they do not want to have to learn to use a new technology, or simply because they are recalcitrant (Joshi, 1991; Lapointe & Rivard, 2005; Marakas & Hornik, 1996). Other researchers suggest that people often resist new technologies for more structural reasons: They learn that taking advantage of the material capabilities provided by a new technology may upset the balance and distribution of roles, responsibilities, and, consequently, existing power relations within the organization (Barley, 1986; Edmondson, Bohmer, & Pisano, 2001; Markus, 1983). By following the premise that meaning is created out of people's everyday interactions, communication researchers have offered a perspective on technology resistance that falls somewhere in between these psychological and structural explanations. Such an approach generally acknowledges that during their initial engagement with a new technology users share information with others about the technology and, in so doing, begin to form interpretations about its functionality (Fulk, 1993; Jian, 2007; Johnson & Rice, 1987; Leonardi, 2003; Poole & DeSanctis, 1990; Rice & Aydin, 1991).

In this article, I follow existing communication research that focuses on how people develop the interpretation that a technology is not good to use and, through their decision to resist it, inadvertently stymie managerially instigated organizational changes of which they themselves are in favor. The ethnographic study reported herein aims to untangle the roles that people's interactions with others, what I term "social interactions," and their interactions with the material features of technological artifacts, what I term "material interactions," play in the process of interpretation formation. The study shows how users of a new computer simulation technology drew on information communicated in their social interactions to develop an interpretation about what the technology should do and how, through their material interactions, they came to decide that the technology was not able to achieve this goal. I then link informants' engagement in social and material interactions to patterns of declining use of the technology and, eventually, to a failed organizational change. I begin by drawing on existing research on communication and technology interpretations to provide a theoretical basis for the inductive study reported herein.

Interpreting technology and organizing work

Borrowing the concept of ‘‘interpretative flexibility’’ from the empirical program of relativism, Pinch and Bijker (1984) were among the first researchers to suggest that the developmental paths of a technology could be explained by differences in interpretations of its meaning. Their framework posited that because different groups associated with a technology’s development held their own goals, objectives, and social constraints, they would each interpret the technology in a unique way. The result of such differing interpretations is that the actual ‘‘functionality’’ of the technology—the criteria by which individuals determine what the technology is supposed to do and by which they evaluate whether or not it does that—varies significantly among the diverse groups associated with it (Bijker, 1995; Pinch, 1996). Thus, interpretative flexibility implies that the meaning of an artifact does not reside in the artifact itself, but is influenced by the ways it is interpreted by relevant social groups. Drawing on such insights, communication researchers began to argue that such a view of interpretation could be extended past explanations of technology development to account for the varied ways in which a technology is used after it is implemented in organizations (Contractor & Eisenberg, 1990; Fulk, Schmitz, & Steinfield, 1990; Johnson & Rice, 1987). Because they are embedded in a social context, technologies can be and often are interpreted in a number of different ways. Indeed, Rice (1987) made the convincing argument that although organizational managers might decide to implement new technologies for what appear to them to be rational reasons (e.g., changes related to technical or economic improvements), users often do not share these goals.

Over the years, a number of communication studies have shown that people develop interpretations of technology by interacting directly with managers and coworkers and talking with them about it (Fulk, Schmitz, & Ryu, 1995; Kraut, Rice, Cool, & Fish, 1998; Orlikowski & Gash, 1994). Fulk et al., for example, proposed that an individual’s perceptions of a technology’s constraints and affordances were formed ‘‘to a substantial degree by the attitudes, statements, and behaviors of coworkers’’ (Fulk, Steinfield, Schmitz, & Power, 1987, p. 532). Their studies of how engineers in a petrochemical company communicated about a new e-mail system demonstrated that social influence processes shaped how individuals perceived e-mail’s affordances and that engineers’ perceptions of the technology were correlated with the perceptions of members of their work groups. Rice and Aydin (1991) were the first to combine social influence and network theories to explore how different sorts of proximal relations (and hence discussions about technology) influenced an individual’s perception of a new medical information system. The data showed that direct communications ties and managers’ perceptions of the technology significantly influenced respondents’ perceived worth of the system, but that their actual use of the system had few effects on their attitudes. They concluded that ‘‘usage of the system, by itself, does not apparently influence one’s attitudes toward the system’’ (p. 238).

Researchers drawing on structuration theory have also focused on how people’s interpretations of a new technology lead them to either accept or reject it and how patterns of use/nonuse encourage or stymie organizational change. Poole and DeSanctis (1992) employed structuration theory in a study of how groups appropriated the features of a group decision support system (GDSS). The system captured all comments that the students entered into their computer terminals, thereby creating a transcript of their interactions. The authors then analyzed the speech acts contained in the transcript using a coding scheme that distinguished among nine types of appropriation moves (DeSanctis & Poole, 1994; Poole & DeSanctis, 1992). The findings indicated that the groups drew on their existing norms for social interaction to develop an interpretation of how the GDSS system should be used. Based on these interpretations only 11 of the 18 groups faithfully appropriated and used the technology in ways that changed patterns of group communication in the directions designers and implementers envisioned. Orlikowski’s (1992) work on the implementation of computer-aided systems engineering (CASE) tools in a consulting firm demonstrated similar dynamics. Orlikowski found that consultants formed interpretations about how the CASE tools should be used through their talk with each other and by reflecting on their existing work practices. As consultants used the technology’s functionality their work began to change. As their work changed, they began to talk about how they were using the technology, and this communication through social interaction, in turn, led them to develop new interpretations about the technology. New interpretations encouraged new uses of the technology, which resulted in further changes in the structuring of work.

Taken together, these studies show that people form interpretations about a new technology as they talk about it with others. These interpretations shape the way that they use the technology, and outcomes of technology use either lead to or stymie organizational change. In recent years, studies of technology interpretations, especially those drawing on structuration theory, have been critiqued for not paying enough attention to the material properties of technologies and to how people actually use them in their work (Chae & Poole, 2005; Orlikowski, 2005; Rose, Jones, & Truex, 2005). To be sure, many organizational communication researchers have focused on how the material properties of technology are implicated in the process of organizing (Hirscheim, 1986; Leonardi, 2007; Lewis & Seibold, 1993; Majchrzak et al., 2000; Manross & Rice, 1986; Nass & Mason, 1990). Most of these studies tend to acknowledge that organizational change occurs at the nexus of people’s communication about a technology and the technology’s material features and seek to explore empirically how people’s use of a technology changes as both talk about the technology and the technology’s features coevolve.

Jackson (1996) has made the strong argument that although a technology’s material properties exist apart from those who use them, until they are interpreted and rationalized by people as being useful those material features can have no effects on the way patterns of communication are organized. As Jackson suggests:

> Functionality—the ability of an artifact to be used to accomplish a social task—is the primary requirement of technology . . . If an artifact is not functional, it will not . . . be even a candidate for attention. Achieving functionality, however, is not a certain feat. To be functional, an artifact must meet two requirements. First, someone must perceive that it can be used to accomplish a particular task, and, second, it must be capable of performing that task. (p. 255)

Orlikowski (2000) has made a similar claim, urging scholars to examine not just what a technology is capable of doing, but examining how technologies are used in practice based on what people interpret them as capable of doing. Yet Orlikowski pushes Jackson's second requirement one step further by claiming that it matters little whether or not a technology is "capable of performing that task" unless people *interpret* the technology as being capable of doing so. Thus, it may be the case that a technology can actually do thing "X," but if people do not interpret the technology as being capable of doing "X" they won't use it, and its material features won't afford them the opportunity to change the way they work. Orlikowski's argument, which has been taken up by several organizational communication researchers (Boczkowski & Lievrouw, 2008; Jian, 2007; Leonardi, 2003), including Jackson (Jackson et al., 2002), suggests that those interested in understanding how interpretations are formed should not just examine what technological artifacts can or cannot do, but should focus instead on how and why people come to believe that they do or do not do those things.

Early work by Johnson and Rice (1987) argued that people's interpretations about (a) what a new technology is supposed to do and (b) whether or not the new technology is able to do, or is efficient at doing that thing arise from talk before it is implemented and immediately subsequent to implementation, and with people's own personal experiences of trial and error as they try to use the technology in conjunction with information systems already employed within the organization. Most organizational change programs in general (cf. Lewis & Seibold, 1998), and attempts at technology-driven organizational change in particular, are surrounded by people's conversations with their managers, coworkers, and others about what the technology is supposed to do (Johnson & Rice, 1987; Majchrzak et al., 2000). Likewise, people spend significant portions of their workday alone attempting to use a new technology and integrate it into their ongoing streams of work (Griffith & Northcraft, 1996; Vaast & Walsham, 2005). Thus, it stands to reason that people develop interpretations of a new technology during their social interactions with others and in their material interactions with the various technologies they use. Leonardi and Barley (2008) observe, however, that there is still a need to understand whether and how people's social and material interactions become intertwined in ways that promote or discourage organizational change.

Drawing on this prior work, I adopt the stance that, in the practice of forming interpretations about a new technology, users act with and in response to each other (social interactions) as they use a new technology, and that they also act with and in response to the material features of technologies themselves (material interactions). What remains unclear, however, is what roles social interactions and material interactions play in the formation of interpretations, whether they have equal influence, and how engagement in these types of interactions can either lead to or stymie planned technology-driven change. I describe the findings of an ethnographic study aimed at uncovering how crashworthiness engineers at a major automobile company formed interpretations of a new information technology called CrashLab. The data suggest that the reasons why engineers interpreted CrashLab in certain ways, and ultimately why they resisted the purposive program of organizational change inscribed in it, cannot be accounted for without considering the relationship between the types of information they constructed through their social and material interactions.

Background and methods

Crashworthiness engineering and technology use

Autoworks (a pseudonym) is a large automobile manufacturer. The vast majority of Autoworks' engineering workforce is located at its technical center in the Midwestern United States and it is at this location that I focused my data collection efforts on the work of Performance Engineers (PEs) who worked in the company's Safety Division doing crashworthiness engineering. The idea behind crashworthiness is that the best chance an occupant has of surviving a crash with little or no injury is for the vehicle to absorb the energy of a collision so that energy does not harm the occupant (DuBois, 2004). PEs evaluate the performance of a vehicle by assessing its ability to protect occupants during a crash.

Because the cost of administering and recording the data for physical crash tests is so egregious, Autoworks requires its PEs to use computer simulations to make recommendations for how the structure of a vehicle can be changed to increase performance in an impact. In an attempt to reduce the time and effort it took PEs to set up and analyze a computer simulation, as well as to standardize the assumptions engineers used throughout the process, technology developers in Autoworks' R&D division built a new technology called CrashLab. CrashLab was a software application that worked in conjunction with existing preprocessing and postprocessing tools.[1] The technology was meant to automate many of the preprocessing and postprocessing tasks that engineers had traditionally performed manually. CrashLab was neither a preprocessing nor a postprocessor; it was a technology that sat on top of existing preprocessing and postprocessing applications already in use by PEs.

By automating much of the preprocessing and postprocessing work of PEs, the makers of CrashLab intended to bring about a significant change in the organization of work in safety. Most modern engineering organizations make a sharp delineation between activities such as model building or drafting, which require technical skill but not detailed engineering intuition and judgment, and analysis activities, which require engineers to possess and apply in-depth domain knowledge (i.e., physics, thermal dynamics) to mathematically complex problems (Bucciarelli, 1994; Henderson, 1998; Vincenti, 1990). In fact, the boundaries that demarcate most formal roles and job responsibilities in engineering organizations, for example, between Design Engineers (DEs) and PEs, are drawn around model building or analysis activities (Suchman, 2000). Participation in analysis activities places an engineering organization in the center of decisions about product architecture and design. Consequently, most engineering organizations seek to increase the amount of analysis work they conduct and reduce the amount of time they spend performing routine simulation building activities.[2] Therefore, shifting the focus of effort from simulation model building to analysis activities marks a significant and highly desirable organizational change in engineering firms.

This desire to change from an organization that spent most of its time building simulation models to an organization that analyzed engineering solutions and provided input to product architecture decisions was very strong within the safety division at Autoworks. Because such a shift would mean that the division could contribute more meaningfully to engineering design decisions, a change in focus from simulation model building to analysis activities would also reward the division with higher status and visibility within the company. Thus, to reduce the amount of time PEs spent building simulation models, and increase the amount of time they spent analyzing them, managers worked with R&D developers to implement CrashLab. As one manager commented:

> We've decided to implement CrashLab so that we can change the way our engineers work. Basically we want the organization to look different. We want engineers to be talking less with each other about routine model building questions and talking more to each other about what the analysis of their simulations means. Basically we want them to change the way they interact. If we can reduce the amount of time it takes them to build a model and then bridge their different assumptions by giving them a common ground, I think we can achieve this goal.

It was to understand how PEs developed interpretations of CrashLab in the practice of their work, and to see how those interpretations would either lead to or stymie the organization's change from a simulation building to analysis focus, that I immersed myself in the world of crashworthiness engineering.

Data collection

I collected data on the work of PEs in the safety division between July 2004 and August 2006. In total, I spent 9 months during this 2-year period conducting observations. CrashLab was implemented and immediately available for use by PEs in the final days of August 2005. I collected 2 months of data (July–August 2004) on the normal work of PEs before they began to use CrashLab and 7 months (August–November 2005, March–April 2006, and August 2006) of data on their actual use of the new technology. One advantage of this data collection strategy was that I was able to capture an *emic* (insider's) understanding of PEs' work prior to their interaction with the new technology. Because I conducted observations during the implementation period I was able to document how informants struggled to make sense of the technology in the real-time practice of their work rather than having to rely on retrospective accounts of their interpretation formation.

During my observations with PEs I compiled a complete record of the actions they took to build and analyze simulations (with or without the use of CrashLab), to correlate those simulations with the results of physical crash tests, and to make recommendations about how to improve the performance of the vehicle in various impact conditions. It quickly became clear that to carry out these actions, PEs needed to spend a lot of time interacting with their coworkers and DEs who created the computer-aided drafting data that were used to build simulation models. PEs also spent much of their time alone at their computers working with various preprocessing and postprocessing applications. Thus, my fieldnotes captured a broad range of solo and group interactions. To capture these actions in detail, my observations with PEs lasted between 3 and 5 hours. Each 3- to 5-hour session was spent observing only one PE as the primary informant, and each PE who participated in this study was observed for at least three sessions. After 10 hours with one informant I was able to understand how the engineer worked and why she worked in that way. In total, I spent slightly more than 500 hours observing PEs at Autoworks.

I captured the interactions that occurred in each observation in a number of ways. I sat behind the informants at their desks while they worked and I followed them when they went to meetings and to talk informally with colleagues. I accompanied PEs to the proving grounds to watch physical crash tests and also went with them to vehicle teardowns, where they were able to inspect the state of the parts after a physical test. During all of these activities I took notes on my small laptop computer, indicating the types of activities the engineers were conducting, why they conducted them, and with whom or what they were interacting. Additionally, I recorded talk occurring during all observations on a digital audio recorder. Using audio recordings allowed me to document the conversations engineers had and to capture their personal thoughts about different matters, which I encouraged them to speak out loud as they worked. I also let the audio recorder run when engineers were working silently at their computers. All of the audio recordings were transcribed verbatim. I later integrated these audio recordings of dialogue with the fieldnotes. By using the digital time stamp on the audio recorder in conjunction with the observation records I was able to document how long informants worked on particular tasks. The combined observation records (fieldnotes with corresponding dialogue) for one observation were between 20 and 30 pages of single-spaced text. The data presented in this article are drawn from 64 observation records made in one group within the safety division—the Piston Group.

Data analysis

I began analyzing the observation data contained in the fieldnotes through a process of selective coding (Strauss & Corbin, 1998) in which I flagged all instances in which PEs talked about CrashLab with someone else, or used its outputs (e.g., plots of energy curves) to inform their conversations. To be counted as an instance, a conversation had to contain at least one of three elements: (a) the involved persons had to explicitly talk about how CrashLab was supposed to change their work, (b) they had to discuss at least one of CrashLab's material features, or (c) they had to use output generated by CrashLab to answer a specific engineering question. By coding conversations for these criteria, three main categories of conversation emerged. I grouped these categories of conversation together under the broad heading of *social interactions* because in each type PEs did not actually use CrashLab's material features. Instead, they talked about CrashLab in a way that was removed from the actual experience of using it.

I read through all of the social interactions that PEs had around CrashLab and used Glaser and Strauss's (1967) constant comparative method for determining what elements were similar among them. By placing codes on topics of conversation that were similar, it became clear that through conversations with managers, customers, and coworkers, PEs were beginning to develop the interpretation that CrashLab was supposed to be a "preprocessing technology." Miles and Huberman (1994) suggest that after forming such an initial empirically grounded hypothesis, the researcher should return to the data and determine if there are more examples that support this initial hunch. Following this method, I returned to the data and sought examples that showed PEs using information from their social interactions to develop the interpretation that CrashLab was "supposed to be" a preprocessor.

Next, I went back through all of the data (selective coding) to uncover instances in which PEs actually sat in front of their workstations and used CrashLab. I flagged all of these instances, many of which spanned multiple pages of the fieldnotes, and read through them. When PEs worked only with the CrashLab technology without any other computer programs open on their desktops (37% of all instances), they did not make many comments to me (sitting behind them and taking notes on what commands they were executing). However, when PEs attempted to use CrashLab in conjunction with another technology (63% of all instances), they often became visibility agitated and would turn to me and make comments about what CrashLab was or was not doing and why such action was or was not problematic. Following Glaser's (1978) recommendation to focus on occurrences in the data where there is a breach from routine practice, I decided to pay close attention to how PEs used CrashLab in conjunction with other technologies. I identified three types of technologies with which PEs normally used CrashLab. I grouped the observed uses of CrashLab in conjunction with these three technologies together under the broad category of *material interaction* because PEs were doing a similar thing in all three cases: Wrestling with the constraints and affordances of the technologies and trying to make CrashLab work with them.

Next, I aimed to uncover whether there was a link between PEs' engagement in social and material interactions and the informal organization of work in the safety division. I constructed a table for each observation record, which I populated with a raw count of the number of times in that observation that an informant engaged in both social and material interactions. Because (as I discuss below) the qualitative analyses suggested that it was through these social interactions that PEs developed an interpretation of CrashLab as a "preprocessor" I took the occurrence of "social interaction" as a proxy for the formation of such interpretations (see, e.g., Becker, Geer, Hughes, & Strauss, 1961). Similarly, I took the occurrence of "material interaction" to represent the formation of interpretations of CrashLab as "inefficient" at preprocessing tasks. Thus, if social and material interactions were indeed contexts in which PEs' interpretations about CrashLab were formed, the frequency with which these types of interactions appeared within each observation should influence PEs willingness to use CrashLab in the future. To construct such a dependent variable, I took the mean score (on a 5-point scale) of informants' responses to three statements: (a) CrashLab is a useful tool for crashworthiness work, (b) I will continue to use CrashLab in the future, and (c) I would recommend that other PEs use CrashLab, to which I asked informants to respond at the conclusion of each session of observation. The mean score of these three highly correlated items ($\alpha = .82$) projected a measure of whether a PE anticipated using CrashLab in the future.

To determine if a relationship did exist between one's engagement in social and material interactions and anticipated future use of CrashLab, I conducted hierarchical linear regression analyses. In the first model, I used the social and material interactions as independent variables. In a second hierarchical model, I included a product term. Because social and material interactions were continuous variables, I converted them to zero-centered variables to avoid problems of multicollinearity. I then analyzed simple slopes following the guidelines suggested by Aiken and West (1991). To test the relationships between these variables over time, I sorted the observation records into two categories that corresponded with my main periods of data collection. The first period represented PEs' initial experiences with CrashLab (September–November 2005) and the second period captured their uses of CrashLab after they had become more familiar with it (March–April and August 2006). In total, 34 observations were included in the first period and 30 observations in the second.

Finally, the changes in the organization of crashworthiness work that developers and implementers hoped CrashLab would engender centered on communicative exchanges among PEs. Specifically, developers and implementers wanted PEs to spend less time communicating with one another about simulation model building activities (those processes that CrashLab automated) and more time communicating about simulation analysis activities. Within the two time periods described earlier, I followed the work of researchers who have used advice-seeking behaviors as a communicative action that constitutes informal structures of organizing (Barley, 1990; Leonardi, 2007; Rice, Collins-Jarvis, & Zydney-Walker, 1999) and counted the number of times in each observation that PEs consulted one another on issues related to building or analyzing simulation models. Regardless of vehicle type or government test, all simulation building required knowledge of at least three activities: (a) how to position a barrier or occupant, (b) where to place accelerometers, and (c) how to define sections. Likewise, to execute a useful analysis, all PEs needed to make decisions based on at least three key factors to improve crashworthiness performance: (a) how to change the geometry of parts, (b) how to change the location of parts, and (c) how to change the materials used to build the parts. Together, these six areas of consultation sketched an outline of whether the communicative exchanges constitutive of the organization of work in the safety division were tied to simulation building or analysis activities and whether the continued use of CrashLab occasioned shifts in the process of organizing. If CrashLab brought changes to the organization of crashworthiness work, the number of consultations about simulation building issues should be higher in the first period than in the second, whereas the number of consultations concerning simulation analysis should be higher in the second period than in the first. To discern whether any differences in consultation patterns existed between the two periods, I employed an analysis of variance test.

Findings

Training and first impressions of CrashLab

In August 2005, 3 weeks before it was implemented and available for use, Autoworks' training center ran a number of classes in which PEs could learn how to use CrashLab. All engineers at Autoworks were required to participate in a certain number of in-service training hours every year as part of a professional development program. Because few courses were specifically geared toward crashworthiness engineers, the CrashLab training generated a very high turnout. A review of the course rosters indicated that of the 66 crashworthiness PEs who worked on simulation analyses, 59 (89%) enrolled in CrashLab training. Because the 8-hour training sessions were conducted in advance of CrashLab's implementation on their individual workstations, PEs learned how to use CrashLab before having the opportunity to actually use it in their work. Trainers walked PEs through the features of the technology and led them through exercises in which they practiced (on mock data) how to setup and analyze simulations.

PEs were quite optimistic about the potential of CrashLab to change their work. As one PE commented after a full day of training: "We're always looking for technologies to make our work easier. CrashLab seems like it could really do that. It seems helpful." Another PE shared similar thoughts: "I like what we saw today about CrashLab. It seems easy to use and it would cut out a lot of my busy work, which would definitely be good." Although most PEs were optimistic about CrashLab, some were more cautious in their evaluations. During the lunch break of one of the training sessions, I overheard two PEs discussing the utility of CrashLab:

PE1: I like it [CrashLab]. It's better than what I thought it would be.
PE2: I don't know. I mean it seems ok but it doesn't seem all that revolutionary.

> **PE1:** That's true, but I don't think it's supposed to be this new big thing (*he raises his hands over his head in emphasis*). I got the impression it's supposed to just cut down on the repetition of doing setup stuff. That would be very helpful.
>
> **PE2:** Yeah, I suppose. That would be good. I'd be real happy if it did that.

As these examples illustrate, PEs were, on the whole, quite hopeful that CrashLab would be a useful tool in their work. Yet, most PEs had to wait more than a month after their training sessions before they could test CrashLab on their own with samples of their routine work. In the following sections, I explore the interactions that PEs had with other people at Autoworks and with technologies used in conjunction with CrashLab.

Social interactions

To evaluate loadcases (conditions through which loads, in the form of energy from an impact, are applied to the vehicle frame) and make recommendations for improved vehicle performance PEs regularly interacted with their managers, coworkers, and customers. My analyses indicate that through their interactions with each of these groups, PEs developed an interpretation that CrashLab was supposed to be used as a technology to preprocess their simulation models. In this section, I describe what sorts of things PEs learned about CrashLab from their interactions with each of these groups.

Managers

The introduction of a new technology is normally surrounded by discourse about what it will do and how it should be used (Leonardi, 2008). At Autoworks, discourse about CrashLab circulated at a number of levels, cascading down from tool developers to engineering managers, and eventually to PEs themselves. Crashworthiness managers were responsible for endorsing new technologies such as CrashLab. Without their approval a new technology could not be implemented. However, these managers did not have the authority to require PEs to use a newly implemented technology. This meant that for developers convincing the managers of the utility of CrashLab was a necessary step to be taken in order to see CrashLab implemented in the user community.

Developers thus framed CrashLab in the way they felt would best attract managers to it: Emphasizing the speed with which it would help PEs build and analyze their models. All vehicle programs at Autoworks followed a standard global vehicle development process. The development process specified a series of milestones, or stage-gates, from conceptual development through production. If any of these stage-gates were missed, the vehicle program could run over budget by millions of dollars, or risk cancellation. Therefore, developers framed CrashLab to managers as a technology that would "speed up" the way PEs worked. Managers who were exposed to this framing then passed such messages on to their PEs. In weekly staff meetings, these crashworthiness managers would use the same talk about "speed" to pitch

CrashLab as a technology that would allow PEs to build their simulation models faster than they had before. As one example, a manager announced to his PEs in a weekly staff meeting:

Manager: So basically we've been asked to evaluate CrashLab again and start using it. From what I know the goal of this tool is to help us speed up the way we get models ready for analysis. That's something we all want, right?

PE: Yeah, that's definitely the boring part.

Managers frequently extolled CrashLab's virtues as a new technology that would help PEs to reduce the time it took to set up models for analysis. Responding to a question from one PE, a manager prefaced the opening to a staff meeting with the following overview of CrashLab:

PE: So will it make the setup procedure faster?

Manager: Yes. After you learn to use it you'll be able to set up models much faster than you do now and there'll be more time to work on other engineering aspects. This will make your work much faster.

For crashworthiness PEs, ‘‘speed’’ was a word associated with preprocessing and not with postprocessing. This was due in large part to the fact that model building activities were considered by most PEs as a necessary but burdensome precondition for analysis activities, where great care and precision were required. Thus, when they heard that CrashLab was supposed to speed up their work, PEs began to liken the technology to preprocessing applications. As one PE commented:

> My guess is that CrashLab is going to be a new preprocessor because it's supposed to make model building faster. That's normally what preprocessors do. Like, you could do it all by hand, but the reason you use a preprocessor is to speed up the work. So, yeah, I guess CrashLab is going to be a good preprocessor if it speeds things up.

Because the only tools that normally ‘‘speed up’’ PEs, work were preprocessors, they began to develop the interpretation that CrashLab itself was a preprocessing technology.

Coworkers

Crashworthiness PEs often talked with coworkers to discern acceptable vehicle designs that would meet the demands of multiple loadcases. During these interactions it was common for PEs to ask questions about the functionality of a particular technology that they employed in their modeling and analysis tasks. This practice is understandable given that once they were through with training PEs did not normally have access to those who trained them. Also, PEs did not normally know or have access to the developers who initially created the technology, and their managers did not have the technical skill to operate it. So if a PE needed to learn about the features of a technology such as CrashLab as she encountered a task for which she hoped the technology would be helpful, her only viable recourse was to ask another PE who had some prior experience using it.

The difficulty that PEs experienced when relying on their coworkers to learn about the capabilities of the new technology was twofold. First, the PE who knew about the functionality of CrashLab, or at least claimed to, did not have much more practical experience using it than the PE who was hoping to learn. Second, a PE who did have experience using CrashLab often felt it was easier (in part because of his own limited knowledge of the technology) to explain the functionality of the technology not in technical terms, but by likening it to other technologies already in the crashworthiness portfolio. As one PE mentioned:

> If someone wants help learning about a tool I'll help. But really, I just tell them what I know. Usually it's easiest for both of us if I can describe what I know about one tool based on what I know or I know they know about another tool. That just makes it easier for everyone. Like in engineering, take what you know about stress calculations and apply that to strain rather than trying to learn about strain by pretending you've never heard about stress. That just isn't practical.

Consequently, when one PE explained the functionality of CrashLab to another, he often used other technologies as benchmarks with which to evaluate whether or not CrashLab could perform the desired activity. At other times, the PE requesting help would benchmark CrashLab against other technologies to try to make sense out of what CrashLab did and did not do well. Consider, for example, the following exchange between PE1 and PE2, who recently completed CrashLab training. PE1 was hoping to learn from PE2 whether CrashLab supported a modular approach to model building through the use of an INCLUDE statement that would call together files containing various modules of a vehicle model and integrate them:

PE 1: Does CrashLab support INCLUDE files?
PE 2: No I don't think so.
PE 1: Are you sure?
PE 2: Uh, I (*pauses*) I took the training just in October and I don't remember that.
PE 1: Oh.
PE 2: Yeah so it's kind of useless for our [vehicle] program.
PE 1: I think most programs are using them [INCLUDE files] now.
PE 2: Well it's probably useless for most programs then.
PE 1: But Hypermesh or Easi-Crash supports them?[3]
PE 2: Yeah.

Objectively, this characterization of the functionality of CrashLab was incorrect. CrashLab did indeed support INCLUDE files. Absent any continued access to developers or trainers, however, PEs were left to their own devices to learn about CrashLab's material features in the practice of use. When they engaged in social interactions characterized by technological benchmarking practices PEs turned to their peers, who were normally not much more knowledgeable than they, for guidance and advice. Therefore, inconsistent and inaccurate information about CrashLab was often perpetuated amongst PEs. The result was that PEs often accepted the incorrect prognosis of another engineer without actually testing the technology themselves.

Instead of evaluating CrashLab on its own merits (as an automator of pre and post-processing functions) PEs explicitly compared CrashLab's features with the features of existing preprocessors and in doing so began to interpret CrashLab itself as a preprocessor. The effects of technological benchmarking encouraged PEs to make determinations about whether or not to use CrashLab based on what they learned it could do or not do in relation to preprocessing tools already available to them. As one PE discussed when asked by a coworker if he should use CrashLab for an upcoming project:

> **PE1:** For me, CrashLab has to do more than Hypermesh for it to be useful. CrashLab is a good niche application I think. It has some good potential to generate mesh for curvatures but the reality is that it's just not as intuitive for me as Hypermesh. So really for me to use a new preprocessor it would have to be significantly better and give me more of an advantage than Hypermesh. That's my benchmark. So I just don't think CrashLab is worth investing the time to learn.
>
> **PE2:** So, it's essentially a preprocessor?
>
> **PE1:** Yup, that's right.

Notice in this quote that the PE considers CrashLab a "new preprocessor." In the eyes of PEs, CrashLab became a "new preprocessor" for "niche applications" like generating "mesh curvatures" rather than as an automator for more advanced pre- and postprocessing functions.

Customers

Through their social interactions with managers and coworkers, crashworthiness PEs began to develop relatively robust interpretations of CrashLab as a preprocessing tool. The social interactions PEs had with the DEs who were their customers gave further validation to their interpretations. PEs and DEs interacted regularly around the results of analyses generated from simulations of various loading conditions. PEs were aware that in order to convince a DE to make changes to the design of his or her part (which most did not want to do because one change cascaded quickly into many more) they had to teach their DE customers about the type of analysis they conducted so that the DEs would understand why it was necessary to redesign the part.

To be effective in promoting a design that bore good performance, PEs had to engage in a fair amount of technical teaching with their DE customers. Consider the following example of technical teaching observed in a routine interaction between a PE and his DE customer:

> **PE:** If we put the bracket there (points to the computer screen to an empty space next to the torque box) it will be in our crush space. [4]

> **DE:** Like I said before, due to our packaging constraints I don't think we can put it anywhere else.[5] I can design it to break away if you need so it won't be a problem. The bracket will still be intact but it will just come loose.
> **PE:** The problem is that if it breaks away the material will still be there and it will stack up.
> **DE:** Stack up?
> **PE:** It will stop us from getting a progressive crush.
> **DE:** Progressive?
> **PE:** Let me show you a picture of what it would look like. Do you have the PowerPoint I sent?

Clearly, this DE, who had worked at Autoworks for many years, was unclear about the specific types of issues that concerned PEs in their analysis. Thus, in the practice of social interaction, the PE had to teach the DE about his analyses so as to convince him why it was necessary to make the particular design change. Due to the complexity of the analyses, the simplest way for a PE to help a DE to understand his need was through pictures. As we see in the above example, this PE, as did most others, brought to their meetings with DEs images (normally screenshots or videos) of the problem areas they encountered in their simulations. As one PE suggested:

> I don't always get what a DE does and I know they don't get what we do. The best way is just to show them an image or video of what's going on. Then they can see what the problem is, they're smart, they see it and then we can talk about the change. If I just brought in the results of my analysis it wouldn't go anywhere. That stuff only means something to crash engineers.

Those PEs who did, on occasion, use CrashLab to automate the postprocessing of their simulations received an HTML report. The difficulty with such a report was that it provided the same postprocessing calculations as the commercial postprocessing technologies already in use in the safety division. The advantage of using CrashLab to automate the postprocessing of a simulation was that by standardizing the filters used to output data and the time-steps at which those filtered data were outputted PEs would have a consistent and concise summary sheet they could use to present to a DE as evidence that a change was necessary. The CrashLab report did not generate images or videos of deformed parts; thus, if PEs used CrashLab for postprocessing automation they still had to open their existing postprocessor and generate pictures or videos to bring to their social interactions with DEs. Because CrashLab's postprocessing functions were neither effective in generating output that existing postprocessors could not, nor was it effective in helping PEs to engage in technical teaching with their customers, PEs continued to interpret CrashLab for what it could, in their eyes, do—preprocessing—and ceased to use CrashLab's postprocessing functions to aid them when interacting with DEs.

Material interactions

In addition to using CrashLab, crashworthiness PEs spent a considerable amount of time interacting with other technologies whose material properties either afforded or constrained their ability to make predictions of the performance of a vehicle for a specific loadcase. Solvers, disk drives, and postprocessors were three kinds of technologies PEs had to use in conjunction with CrashLab in order to carry out their simulation analyses. In this section, I describe how interacting with these other technologies led PEs to determine that CrashLab was not effective as a preprocessor.

Solvers

PEs used a preprocessing technology to convert computer-aided drafting geometry into a simulation model. Once the model was built, it was written to a simple text-based file called an "input deck." These input decks were then submitted to a solver—a program on a supercomputer that applies equilibrium equations to the model and solves these equations for unknown values using linear or nonlinear numerical schemes (Hughes, 2000). In effect, the graphical interface of a preprocessing technology, or of CrashLab, was for user convenience only. Because PEs worked in a virtual, math-based world, the input decks that preprocessing technologies created sometimes contained arrangements that could not exist in the physical world. For example, a particular part may have had no mass associated with it or two parts might have intersected. Such errors in an input deck, if they were minor, would cause the solver to produce results that were inaccurate. If the errors were large, the solver would be unable to compute the particular loadcase and the job they submitted would, as informants said, "bomb out." Input decks were so large that it was nearly impossible for a PE to review the entire file to look for errors in the model. Therefore, PEs normally submitted their model to the solver for a 0-millisecond run. This meant that the solver would attempt to compute the model before the first time-step. If the solver discovered problems, such as intersecting parts, it would return an error message. The PE could then use this message to preemptively debug the input deck before submitting it for a full run.

An analysis of my fieldnotes indicated that PEs spent roughly 17% of their time debugging models that bombed-out from 0-millisecond runs. As indicated earlier, this debugging did not necessarily indicate that things could have been done better before. Instead, preemptive debugging was expected because of the complexity of the models with which PEs worked. The trick to effectively debugging a model was that the PE had to be able to translate the error message generated by the solver into an understanding of where to look in the input deck to correct the problem. When using CrashLab to prepare a model for submission to the solver, PEs quickly discovered that CrashLab formatted the input deck in a way that was different from what they were accustomed to seeing. In fact, CrashLab was automating a number of the processes the PEs used to set up their models and they were unclear on how such modifications were displayed in the input deck. If one considers that PEs spent, on average, 17% of their time engaged in preemptive debugging activities, their frustration that CrashLab produced an input deck they could not understand was quite warranted.

Additionally, PEs worried that by not understanding the contents of that input deck they were abdicating control to CrashLab—control that they believed was in their best interest to maintain because it allowed them to have confidence in their results:

> When Hypermesh writes a [input] deck, or any preprocessor, it puts all this extra junk in. So what you have to do is go through and clean it up before you submit it. The other issue is that let's say you use a tool like CrashLab and it automatically generates your deck for you. Well you don't know how it's doing it or what it's putting in there. So I guess you've lost some control over the process. This becomes a real issue if you have to debug it because it bombed-out. If you didn't know what was in the deck in the first place and didn't control your output yourself you won't know where to begin debugging it and it could take you forever. I don't have forever; I don't even have an hour.

Because the material features of CrashLab acted on their input deck in ways they could not understand, PEs grew afraid that using CrashLab would make preemptive debugging more complicated than it was with other preprocessing tools. Thus, PEs began to interpret CrashLab as an inefficient technology for preprocessing.

Disk drives

Crashworthiness PEs used full-vehicle models (math-based models where mostly every component in the vehicle is represented) in their work because nearly every part in the vehicle is implicated in an impact condition. If one considers that PEs normally conducted upward of 30 iterations on each of these models, it becomes very clear that data storage space was a major issue for crashworthiness PEs.

PEs had several drives available to them on which to store their data. Their desktop CPUs provided storage space, but were not backed up, so PEs tried not to store important files there. PEs also had a storage space hosted on a backed-up server, but the size of this drive was limited procedurally by the IT department. Finally, the supercomputing center allotted each engineer a certain amount of space on another shared server called the R-drive. The R-drive was where the supercomputing center returned the models to PEs once they had been solved. If an engineer was out of space on the R-drive, her model would not be returned. More importantly, if the combined size of the jobs submitted to the solver was equal to the remaining space on a PE's R-drive, the supercomputing center would not let the PE submit additional jobs. In short, to be able to submit new jobs PEs had to be sure to clean out their R-drive and transfer the returned models to one of their other drives. Based on the average model size, PEs could store only about seven returned models on the R-drive. In effect, this limited them to submitting seven iterations at a time. As we saw earlier, however, most analyses required nearly 20 iterations to optimize a design.

To deal with this problem, PEs looked for ways to subvert the system and submit more jobs simultaneously. The most frequent practice used by PEs was to borrow someone else's submission password so the model would return to that person's R-drive. They referred to this practice as "dumping data" into someone else's drive. The following exchange illustrates the prevalent role that data-dumping practices played in PEs' work:

PE 1: Are you running any jobs?
PE 2: Yes.
PE 1: How many?
PE 2: Two, why?
PE 1: I'm just looking.
PE 2: Looking for someone to submit for you?
PE 1: Yes (laughs), we submitted so many we ran out of space. Even counting six at a time we ran out of job spots.
PE 2: After 6:00 they'll probably go in.
PE 1: But what I'm saying is I've got more than 24.
PE 2: Oh, I see you're looking for IDs to run stuff on. Now I see what you're asking for.
PE 1: Yeah.
PE 2: I borrowed Javier's password to do this the other day.
PE 1: Oh you did?
PE 2: Yeah, but I don't remember it.
PE 1: Are you using Regina's?
PE 2: Am I using Regina's? No. I know Clark was using it I'm pretty sure.

The practice of data dumping was pervasive among crashworthiness PEs. Not only did PEs use it when looking to submit jobs to the solver, but they also used it when their other two drives were full and they wanted to move data from the R-drive and temporarily hold it in someone else's storage space.

Not surprisingly, PEs were very concerned about any technologies or practices that would add extra data to their already maxed-out drives. As discussed earlier, PEs engaged in a number of activities that led them to believe CrashLab would amplify their data storage woes. Through technological benchmarking PEs (inaccurately) learned that CrashLab did not support INCLUDE files, suggesting to them that the types of models they had to submit were going to be larger than when they used other programs that did support an INCLUDE format. As one PE commented:

> If you want to use CrashLab you have to submit one large file, it doesn't support INCLUDE files. So this means every job that comes back would be this huge file I'd have to put somewhere. I have so many space issues that I don't want to deal with that. I think it just helps to relieve my space woes to stick with the preprocessor I'm using now. At least I know how to manage these issues with it.

PEs who used CrashLab's HTML report generation function had to produce one large report for each run they submitted to the solver. As one PE observed, if a full report was generated for each iteration of a model containing information that was not completely necessary for the particular analysis the PE wished to carry out, she would have to store that report somewhere too:

> I don't know what I'd do with all those reports. Those are big files. I don't have any place to store them and I don't know if I'd need them. I think if I just used something like eCrush (a script for postprocessing) I can find what I need and don't have to do something with a big report. It's just not very efficient to use it I guess over like Hypermesh where you don't get those big files.

Because PEs frequently struggled with data management and engaged in the practice of data dumping, they quickly came to view CrashLab as an inefficient technology in terms of their space requirements as compared to existing technologies already in use in safety.

Postprocessors

Iterative testing was quite common in crashworthiness engineering work and happened at a much more dramatic scale, as a part's location in a load path became more central. As one PE recounted to the researcher:

> I don't think our customers understand how much iteration we do. I mean I run like five studies to try to optimize the gauge of the front rails. Each of those studies has six variations to them. That's 30 iterations I'm running and that's just one set of the maybe 20 studies I do for frontal ODB.

Iteratively testing the deformation of parts marked a key component of crashworthiness engineering analysis. In fact, it was common for PEs to run 30 iterations for one part in order to verify that its virtual performance was indeed representative of its performance in a physical test and to optimize on a certain design with the goal of achieving the best performance possible given a set of initial constraints.

After completing a number of iterative changes to a part, a PE would postprocess the runs based on the particular performance variable of interest (i.e., stress, energy, intrusion). To make sense out of the large number of iterations, and ultimately to determine which iteration provided the best performance on that particular metric, PEs usually plotted the results of the various iterations against one another in a graph. To produce such graphs, PEs typically used a postprocessing application called Hypergraph. Because the solver outputted the results file in a text-based format, Hypergraph could easily read in the results. However, a PE had to run each iteration as a separate test and then aggregate the results at the end. It was not until all iterations had been solved (which could take up to a week) that the PE had a sense of how the results compared to one another and he or she could begin to make suggestions for changes to a part.

Unlike the other technologies used by PEs, CrashLab outputted its results in HTML format (in large part to provide a "clean" document PEs could use to present results to their customers). But Hypergraph could not read HTML data, so PEs had to manually extract the data generated in the CrashLab HTML report and type that data into the postprocessor. This added an extra step to PEs' work—a step that took more time to complete—and by requiring manual data entry introduced more opportunities for PEs to make errors. As one PE noted: "It's [using CrashLab] just not an efficient way to work because you have to go through extra steps to get your results, and I'll probably screw it up!" This engineer and many others noted that CrashLab did not function well in conjunction with postprocessors and, thus, was not a very effective preprocessing application.

Reducing use of CrashLab and stymieing organizational change

The foregoing data demonstrate that PEs formed interpretations about CrashLab when they talked about the technology with their managers, coworkers, and customers (social interactions), and when they tried to use CrashLab in conjunction with solvers, disk drives, and postprocessors already in use at Autoworks (material interactions). My analysis shows that through their social interactions, PEs developed an interpretation that CrashLab was supposed to be a preprocessing technology. Through their material interactions, PEs developed the belief that, as a preprocessor, CrashLab was inefficient at conducting model setup tasks. At the end of my observations with PEs (50 weeks after CrashLab was implemented), I conducted one-on-one interviews with all of the informants who I observed. I asked them directly: "After nearly a year since it was implemented, what do you think of CrashLab today?" The following answers were representative of the majority position:

> CrashLab just isn't very good at model setup.
>
> I stopped using CrashLab because it didn't work as good as other preprocessors.
>
> CrashLab was ok, but Hypermesh and other preprocessors are better.
>
> CrashLab was supposed to make setup easier, but it actually made it harder to do in lots of cases.

One informant provided the following explanation for why he decided that CrashLab was an inefficient preprocessor:

> When I was talking to people about CrashLab I was always trying to find out more about it. People would tell me some things about it and they all sounded fine. I think most of us are the kind of people that are open minded to new technologies. So if it sounds like it could be helpful you might just try it—at least you definitely keep an open mind about it. You don't really get a good opinion about whether it's good by just talking to other people; you have to use it yourself. So like for CrashLab, I tried to use it to do some mesh generation and it was ok, but then when I tried to use it to write input decks it was a mess. So I guess it was when you use it that you decide if it's going to be good even if you were open minded before that.

In 87% of all interviews, informants referred to CrashLab as a "preprocessor." In 93% of those cases, informants made some negative comment about CrashLab's ability to function as a preprocessor. Most informants indicated that they would not continue to use CrashLab because it was inefficient at preprocessing tasks.

The qualitative analysis presented earlier illustrated how PEs' social interactions led them to develop interpretations that CrashLab was "supposed to be" a preprocessor and their material interactions technologies led them to believe that, as a preprocessor, CrashLab was an inefficient technology. Thus although PEs were interacting with people and with complementary technologies at roughly the same time, the analysis suggests that to develop the interpretation of CrashLab as "an inefficient preprocessor," PEs had to first interpret it as a "preprocessor" before they could interpret is as "inefficient." In other words, PEs discussions about CrashLab with managers, coworkers, and customers had to outpace, at least initially, their attempts to use CrashLab in conjunction with solvers, disk drives, and postprocessors. Figure 1 displays the average frequency that social and material interactions appeared in the data by month. As the figure indicates, PEs were talking with people about CrashLab more in the first 3 months than they were actually using CrashLab in conjunction with other technologies. Although there was certainly overlap between these social and material interactions, PEs' social interactions were dominant in their early interpretation formation.

The results displayed in Figure 1 suggest, along with the qualitative data presented earlier, that PEs' interactions with other technologies served to qualify the interpretation of CrashLab as a preprocessor that they were already beginning to develop. In other words, it seems that, at least initially, PEs' social interactions did little to provide them with information about whether PEs thought that CrashLab was a "good" or "bad" preprocessor. It was only through their material interactions

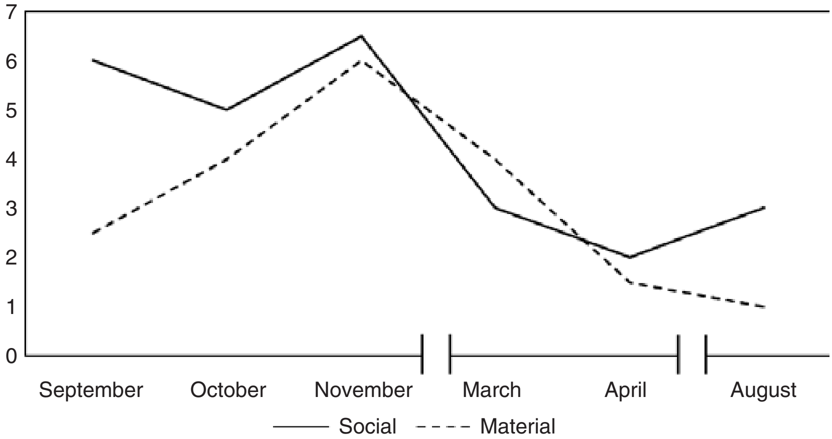

Figure 1 Average number of social and material interactions conducted per observation (*Note*: Double slash marks on *x*-axis indicate breaks in data collection).

that PEs began to interpret CrashLab as being inefficient at preprocessing tasks. In more quantitative terms, PEs' material interactions may have moderated the effects of their social interactions. In their final interviews with me, PEs often made such claims:

> I knew that CrashLab was supposed to be for model setup type work but after I did some preprocessing with it for awhile I started to figure that it wasn't very good for my needs because it generated files that were too big and it just seemed that it wasn't as good as other preprocessors. So basically after awhile I started thinking that I shouldn't use CrashLab anymore if I had to do that kind of work.

To corroborate the assertion that PEs' engagement in material interactions moderated their social interactions, I ran a hierarchical regression analysis using the number of times that PEs engaged in social interactions as one independent variable, and the number of times PEs engaged in material interactions as a second independent variable. In the first block, I regressed the mean deviations of these two independent variables against PEs' rankings of how likely they were to use CrashLab in the future. In the second block, I included the product of the mean deviations of PEs' social and material interactions to test for an interaction effect.

The results of the hierarchical linear regression procedure are presented in Table 1. The regression equation with social and material interactions as separate independent variables (block 1) was statistically significant, $F(2, 61)= 16.44$, $p < .001$, $R^2 = .35$, indicating that PEs' participation in social interactions was positively related to their willingness to continue using CrashLab. Participation in material interactions was negatively related to their willingness to use CrashLab. The qualitative data help to explain the nature of these relationships. PEs began to learn about what the technology could do from their interactions with managers, coworkers, and customers. As many PEs commented, they were generally open minded about CrashLab and most were interested to try it out on their own. Thus, in just talking about CrashLab with others and hearing about what it could do, PEs were likely to be willing to use it in the future. However, when PEs actually used CrashLab in their work, their material interactions prompted them to believe that it was inefficient and, thus, less likely to want to use it again.

The second regression model (block 2), which included the product term, was also statistically significant, $F(3,60)= 13.31$, $p < .001$, $R^2 = .40$. Both social and material interactions were significantly related to CrashLab use. More importantly, however, the results indicate that the interaction effect was significant, suggesting that material interactions moderated the effects of social interactions on CrashLab use. The negative beta coefficient of the product term suggests that as the number of material interactions increased, the effect of social interactions on PEs' willingness to use it in the future declined. Simple slopes analysis indicated a negative association between material interaction and anticipated use of CrashLab across all values of social interaction. However, the negative slopes became increasingly steep as the amount of social interaction increased (social interaction low: $b = -6.92$; social

Table 1 Hierarchical Regression Analysis for Anticipated Future CrashLab Use

	Variables	Beta	Partial Correlation
Entire dataset ($n = 64$)	Block one		
	Social interactions	.33 (3.20)**	.38
	Material interactions	−.49 (−4.76)***	−.52
	Block two		
	Social interactions	.25 (2.29)*	.28
	Material interactions	−.54 (−5.29)***	−.56
	Social × Material interactions	−.24 (−2.22)*	−.28
Period 1 ($n = 34$)	Block one		
	Social interactions	.65 (5.85)***	.72
	Material interactions	−.37 (−3.56)**	−.52
	Block two		
	Social interactions	.53 (4.50)***	.64
	Material interactions	−.46 (−4.09)***	−.60
	Social × Material interactions	−.26 (−2.17)*	−.37
Period 2 ($n = 30$)	Block one		
	Social interactions	−.63 (−3.82)***	−.59
	Material interactions	−.12 (−.73)	−.14
	Block two		
	Social interactions	−.72 (−5.17)***	−.72
	Material interactions	−.07 (−.49)	−.09
	Social × Material interactions	.44 (3.60)***	.58

Note: Numbers in parentheses are t tests for corresponding parameters.
$^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$.

interaction at the mean: b = −11.5; social interaction high: b = −16.26). Paired with the qualitative data presented earlier, this result can be interpreted to show that as PEs engaged in more material interactions they began to think that CrashLab was inefficient at preprocessing tasks. The belief that CrashLab was an inefficient preprocessor made it likely that they would choose not to use it in the future.

These quantitative findings complement that qualitative data presented earlier by demonstrating that when the PEs learned through their social interactions that CrashLab was supposed to be a preprocessor, they were inclined to use it in the future. As they engaged in material interactions and determined the technology was not good at doing preprocessing tasks, the likelihood that they would continue to use it diminished. What the foregoing quantitative analyses do not take into account, however, is the character of the relationship between social and material interactions over time. To explore the effects of time, I conducted separate hierarchical regressions for each of the two time periods. The results from the first time period (Table 1) show a very similar trend to the results for the test conducted on the entire dataset. During this first period, the model that took into account the effects of social and material interactions on future use of CrashLab was significant, $F(2,31)= 25.95$, $p < .001$, $R^2 = .63$, with a positive beta coefficient for social interactions and a negative beta coefficient for material interactions. The second block, $F(3,30)= 20.95$, $p < .001$, $R^2 = .68$, also produced a significant interaction effect with a negative beta coefficient. The simple slopes analysis indicated that between September and November engagement in a high number of material interactions moderated the effect that social interactions had on future CrashLab use (social interaction low: $b = -4.92$; social interaction at the mean: $b = -9.41$; social interaction high: $b = -13.9$). As the analysis shows, the negative slopes became increasingly steep as the amount of social interaction increased.

The results for the second time period (Table 1), which included March, April, and August, show an interesting shift in the relationship between social interactions, material interactions, and future use of the technology. In block one, $F(2,27)= 12.88$, $p < .001$, $R^2 = .49$, both the social and material interaction variables have negative beta coefficients and only the social interaction variable is a significant predictor of future CrashLab use. In block 2, $F(3,26)= 16.72$, $p < .001$, $R^2 = .66$, the interaction term is significant with a positive beta coefficient, whereas the percentage of variance explained jumps by 17%. Simple slopes analysis indicated a negative association between material interaction and anticipated use of CrashLab when social interactions were low ($b = -10.3$), a slope close to zero at the mean of social interaction ($b = 1.62$), and positive when social interaction was high ($b = 13.54$). These analyses suggest that if a PE engaged in even the slightest number of material interactions, the effects of social interaction on CrashLab use were dramatically moderated. The negative statistically significant beta coefficient of the social interaction variable in both models suggests that during the second time period, PEs were starting to circulate information with each other that CrashLab was not just a preprocessor, but that it was inefficient, too—information that they learned through their material interactions in period 1. As the qualitative analyses suggest, this information alone, apart from any participation in material interactions, was enough to make PEs wary of using CrashLab in period 2. However, once PEs who learned from others that

CrashLab was an inefficient preprocessor and then had experiences trying to use it in conjunction with other technologies (material interaction) they were even less likely to use it in the future. As one PE commented in April:

> I've been hearing things about CrashLab for a while. I used to hear it was for preprocessing and no one had much of an opinion on it. But lately I've been hearing real, you know, negative things about it. I decided to give it a try to see for myself 'cause, you know, some people are just whiners, and let's just say I after using it my experience made me believe they're right.

Thus, it seems that PEs' interpretations of CrashLab as an inefficient preprocessor were crystallized within the first 3 months of using the new technology and it is in this time period that PEs decided whether or not CrashLab was worth future use. By month 6, the tenor of social interactions changed from neutral to negative, and engagement in social interactions alone was sufficient to stop PEs from using CrashLab in the future, though corroboration through material interactions dramatically strengthened one's opinions of the technology's inutility. CrashLab had become taken-for-granted as an inefficient preprocessor and PEs had decided to not use it (at least in any substantive way) for future work.

Having found that material interactions moderated the effects of social interactions on future CrashLab use and that, over time, PEs were using the technology less frequently (Figure 1), it seemed unlikely that CrashLab could have brought about the important changes to engineering work of which developers, managers, and PEs themselves were strongly in favor. This is because those desired changes to the organization of work were dependent upon PEs actually using the technology. Based on a dynamic view of social structure, the actions and interactions of individuals are constitutive features of the organizing process (Giddens, 1984; Goffman, 1983; Strauss, 1978). Thus, when PEs communicated with each other about simulation building and analysis activities, they were actively participating in the organizing process. Following such logic, a determination of whether PEs' reduction in the use of CrashLab stymied organizational change can be obtained by measuring the differences in frequency of communicative exchanges revolving around simulation building or analysis activity in each of the two time periods.

If CrashLab did bring changes to the informal interactions constituting the organization of crashworthiness work, PEs would have engaged in more exchanges about simulation building in period 1 than in period 2 while the number of exchanges about simulation analysis would have been greater in period 2 than in period 1. Analysis of variance tests indicate no significant differences in the average frequency of exchanges related to simulation building, $F\ (1,62) = 2.93$, $p =$ n.s., or analysis activities, $F\ (1,62) = .03$, $p =$ n.s. Thus, it appears that by decreasing their use of CrashLab, over time, PEs stymied the significant organizational change developers and managers hoped it would bring about in crashworthiness work. Indeed, throughout my tenure at Autoworks, I did not notice that PEs spent a significant percentage of their work day talking with one another about how to optimize their models and do better analysis. PEs were aware that they spent the majority of their consultations with one another discussing issues of model building as opposed to analysis and that they often commented that they did not see a change on the horizon. One PE's unsolicited comments to me in August 2006 illustrate this point:

> One of the key things they [management] keep telling us is that we should be doing more analysis, more iterations. You know, they say we should collaborate more on coming up with better design. But how are we supposed to collaborate more. We don't have time to discuss all the details of a model because we're too busy building the models. If they want that to happen they need to outsource the model building or automate it so we don't have to do it. Honestly I'd be real excited if they did either one. Send it to India or just get some technology to help.

Another PE made a similar comment when asked by his manager in a staff meeting in August why he hadn't discussed his findings for a pedestrian protection analysis with a more senior engineer:

> The way things are now I don't think that's really feasible. Who's got time to talk about their findings and troubleshoot their analysis? I don't. I mean I don't think anybody does and at least for the foreseeable future I don't think that's going to change.

Thus as another engineer predicted: "We're going to be a basic model building shop for awhile instead of doing real design and analysis work."

Discussion

The combined qualitative and quantitative analyses presented herein begin to paint a picture of how social and material interactions, interpretation formation, technology use, and inadvertent rejection of organizational change might be linked. The findings suggest that through their social interactions with managers, coworkers, and customers, PEs developed an interpretation that CrashLab was "supposed to be" a technology for preprocessing simulation models. Through their material interactions with solvers, disk drives, and postprocessors (all technologies used in conjunction with CrashLab), engineers developed the interpretation that, as a preprocessor, CrashLab was an inefficient technology. The recognition that CrashLab was (a) supposed to be a preprocessor and (b) not good at doing preprocessing work led engineers to make the principled decision to resist using it. Interestingly, by resisting it, PEs could not take advantage of CrashLab's capabilities to speed up their model building work (an objective functionality of the technology) and therefore could not realize an important organizational change that they themselves were very much in favor of: Shifting the concentration of their communication activities from queries about model building to in-depth discussions about model analysis.

Crashworthiness PEs began developing interpretations of what kind of technology CrashLab was through their interactions with managers, coworkers, and customers.

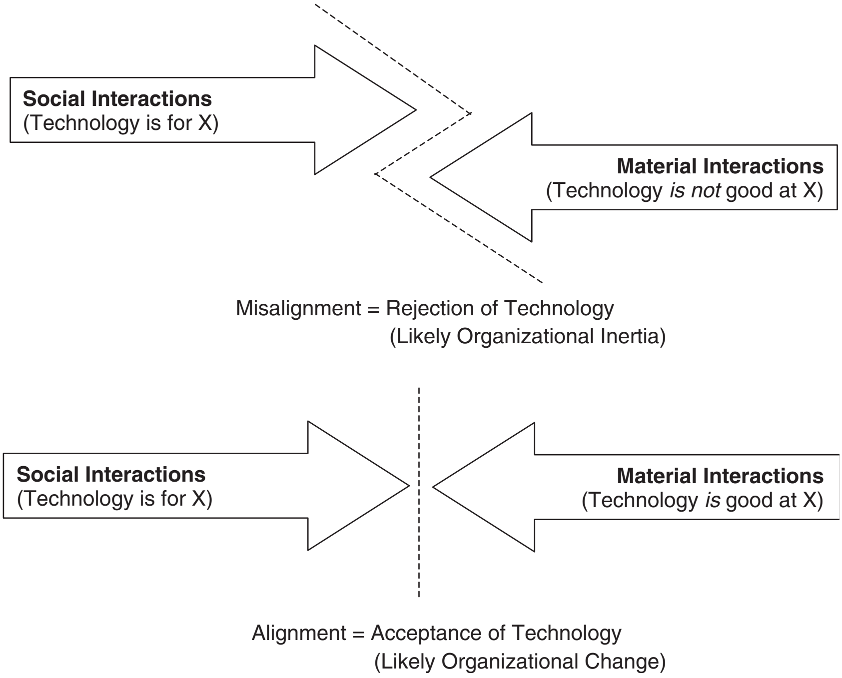

Figure 2 Misalignment and alignment between social and material interactions.

PEs took these understandings emerging from their social interactions and used them as a rubric with which to evaluate CrashLab's efficacy as they engaged in their material interactions. PEs encountered a misalignment between the interpretations emerging out of their social interactions, on the one hand, and their material interactions, on the other. By misalignments I mean that one's engagement in social interactions produces an understanding that the new technology should do a certain thing (X) and one's engagement in material interactions produces an understanding that, as a technology meant to do X, it is deficient in important ways. Figure 2 presents two hypothetical scenarios demonstrating misalignment and alignment between social and material interactions. If social interactions with colleagues and customers produce an understanding that a new technology should do X and material interactions with both new and existing technologies negatively qualify (show it is not good/efficient/proficient at doing X) the information generated in those social interactions it is likely, as we saw in the case of PEs' use of CrashLab, that the new technology will be rejected. As a potential driver of novel communicative interactions, rejection of the technology may very well lead users to resist change and promote inertia in current forms of organizing, as it did at Autoworks. Thus, when PEs rejected CrashLab they did not do so because they were recalcitrant (Marakas & Hornik, 1996) afraid of new routines (Martinko, Henry, & Zmud, 1996), or even because they risked losing status and power (Lapointe & Rivard, 2005); instead, they rejected

CrashLab because they had made a principled (and well-reasoned) decision that using the technology as a preprocessor would impede them from working effectively.

Casting PEs' interpretations of CrashLab as an inefficient preprocessor as "misguided" or even "wrong" threatens to miss the important point that PEs acted as if their interpretations were indeed accurate. By rejecting CrashLab because it was, as one PE commented, "just not a very good preprocessor, by comparison," PEs inadvertently stymied the important organizational changes that managers and implementers hoped the technology would bring to their work. PEs were, by all accounts, excited at the prospect of doing less simulation model-building work and more analysis work. They were eager to spend more time discussing "real engineering" (analysis) issues with their colleagues instead of issues pertaining to what one informant called "monkey work" (simulation model building). Thus, in this case, PEs did not resist organizational change because they felt it was forced on them by management and not within their best interest, as many studies would suggest (Joshi, 1991; Robey & Sahay, 1996); rather, they rejected it because misalignments existed between the information they received (or created) about CrashLab in their social and their material interactions.

It is possible, however, that alignment can exist between the information about a technology that people generate in their social and material interactions. In such a scenario social interactions produce an understanding that a technology should do X, and material interactions positively qualify that technology as being quite capable at doing X. Because many other studies have demonstrated that if people perceive a new technology to be proficient at accomplishing certain goals and/or better than the technologies that they currently use, they will likely adopt the new technology in ways that allow them to change their work (Johnson & Rice, 1987; Leonardi, 2007; Rice & Shook, 1990). In such a scenario, we would expect users to actively engage with the new technology and communicate in ways that may lead to changes in the process of organizing. Majchrzak et al. (2000), for example, found that alignment between the features of a collaboration technology and the goals of the distributed team who used it was essential to allow team members to build a prototype rocket. It was only after the team members adapted the technology and their patterns of interaction to fit each other's demands that they were able to organize their work in such a way to allow them to achieve their work-related goals. In some cases, to be able to align interpretations with a technology may simply mean changing the nature of people's social interactions (their talk about the technology) whereas in others, it may mean altering the social structure in which users are embedded or the material features of the technologies they use (Lewis & Seibold, 1993; Rice & Rogers, 1980).

Consequently, an important implication for the management of new technology implementation is to be mindful that even though the way people talk about a new technology may not entirely dictate the kinds of interpretations they form about it, talk provides a frame of reference that will be qualified (either positively or negatively) by one's material interactions. Had PEs learned through their social interactions that, for example, CrashLab was supposed to be a postprocessor, their interactions with solvers and disk drives might not have stood in their way of viewing CrashLab as efficient for postprocessing tasks; it is possible that PEs would have qualified CrashLab, as a postprocessor, as a fully capable and important tool. Managers and implementers may be wise to attend to the social interaction environment into which a new technology is implemented. Research has shown that although users normally talk to one another to discern information about a new technology (Fulk et al., 1995; Kraut et al., 1998; Rice & Aydin, 1991), managerial explanations about the nature of technological change often overshadow information obtained from colleagues (Leonard-Barton & Deschamps, 1988). Therefore, managers and implementers may wish to introduce information into users' contexts of social interaction that aligns with the experiences those users will have when they engage in material interactions. In short, explanations of people's technology use that only focus on the social interactions out which they form their interpretations of its functionality may be incomplete. Materiality matters in how people form interpretations about a new technology. As this study has shown, if the information users glean from their material interactions negatively qualifies the information they generate in their social interactions, they will stop using the technology and may unwittingly resist even those organizational changes of which they are in favor.

These insights have important implications for theory on the role of technology interpretations in organizational change. At a granular level, the data indicate that interpretations may form early after the implementation of a new technology. Other studies have suggested that most major organizational changes occurring after technology implementation happen within the first 6 months (Leonardi, 2007; Majchrzak et al., 2000). Similarly Johnson and Rice (1987) showed that people's initial interpretations of new word-processing technologies were powerful in predicting organizational outcomes that occurred much later. In this study, PEs interpretations of CrashLab as an inefficient preprocessor seemed to crystallize in about 3 months after it was loaded onto PEs, workstations. Given that the quantity of PEs' social interactions about CrashLab was initially greater than the quantity of their material interactions, it seems that talk about new technology is extremely important. Managers and other invested parties may have a window of opportunity to enter into the communication environment, enroll stakeholders, and begin to shape communication about a new technology that will align this discourse with what people will experience in their material interactions (Lewis & Seibold, 1996). The ability to do so is, of course, predicated on the assumption that managers understand the nature of subordinates' work and that developers understand how a new technology will interact with other tools already used in the organization. Without this knowledge, it would be difficult to influence discourse in such a way that alignment occurs.

At a more general level, this study has implications for theorizing about technology and organizing. The findings showed that people's interactions with the materiality of artifacts are just as important in explaining how they organize their work as are the interactions they have with other people. For many years, apart from a handful of pioneering studies (Johnson & Rice, 1987; Majchrzak et al., 2000; Poole & DeSanctis, 1992), most empirical research adopting social influence or structurational perspectives downplayed materiality's role in the process of organizing. It is possible that this tendency can be traced to researchers' desires to counter deterministic claims. Early theorists married materiality with a deterministic ontology, claiming that features of an organization's core technology determined the optimal way to organize work (Leonardi & Barley, 2008). Despite the fact that the technologies (large-scale manufacturing systems) discussed by many early organizational theorists were quite dissimilar from the information technology artifacts most researchers study today, the equation of materiality with deterministic thinking persists in much organizational research. Although the tenets of determinism may very well be incommensurable within the voluntaristic ontology advocated by most social constructivist approaches, studies of technology's materiality need not be. The findings of this investigation have shown that people's social and material interactions do not exist on orthogonal planes; instead, materiality moderates the effects of communication on people's interpretation formation. Thus, as Rice and Gattiker (2001) suggest, paying closer attention to how people's interactions with and interpretations of the material features of technologies become sublimated into organizational structures may be one way for communication researchers to advance theory on the role of technology in organizational dynamics.

Acknowledgments

I thank Steve Barley, Diane Bailey, and Jake Harwood for helpful comments on previous versions of this article. Earlier drafts of this article were presented at the 2008 meetings of the International Communication Association in Montreal and the Academy of Management in Anaheim. This article was submitted as a representative piece from my dissertation for the 2008 Gerald R. Miller Award from the National Communication Association, and was selected as a winner.

Notes

1 A preprocessor is a software application that allows an engineer to build a computer simulation and a postprocessor is a software application that allows him/her to analyze its results.

2 This shift is often reflected most prominently in an organization's decision to outsource simulation building or drafting work, thereby freeing up more resources in-house for engineers to do analysis.

3 Hypermesh and Easi-Crash are both preprocessors commonly used at Autoworks.

4 The term "crush space" refers to an empty space in a vehicle's load path into which the vehicle's structure can deform. A "torque box" is a structural feature located between the engine and the firewall.

5 The term "packaging constraint" is used to refer to the available space to locate a part in the vehicle. All vehicle parts must "fit" within the vehicle's architecture and are thus packaged according to a master blueprint specified by a vehicle's lead program manager.

References

Aiken, L., & West, S. (1991). *Multiple regression: Testing and interpreting interactions*. London: Sage.

Barley, S. R. (1986). Technology as an occasion for structuring: Evidence from observations of CT scanners and the social order of radiology departments. *Administrative Science Quarterly*, **31**(1), 78–108.

Barley, S. R. (1990). The alignment of technology and structure through roles and networks. *Administrative Science Quarterly*, **35**(1), 61–103.

Becker, H. S., Geer, B., Hughes, E. C., & Strauss, A. L. (1961). *Boys in white: Student culture in medical school*. Chicago: University of Chicago Press.

Bijker, W. E. (1995). *Of bicycles, bakelites, and bulbs: Toward a theory of sociotechnical change*. Cambridge, MA: MIT Press.

Boczkowski, P. J., & Lievrouw, L. A. (2008). Bridging STS and communication studies: Scholarship on media and information technologies. In E. Hackett, O. Amsterdamska, M. Lynch, & J. Wajcman (Eds.), *New handbook of science, technology and society* (pp. 949–977). Cambridge, MA: MIT Press.

Bucciarelli, L. L. (1994). *Designing engineers*. Cambridge, MA: MIT Press.

Chae, B., & Poole, M. S. (2005). The surface of emergence in systems development: Agency, institutions, and large-scale information systems. *European Journal of Information Systems*, **14**, 19–36.

Contractor, N. S., & Eisenberg, E. M. (1990). Communication networks and new media in organizations. In J. Fulk & C. Steinfield (Eds.), *Organizations and communication technology* (pp. 143–172). Newbury Park, CA: Sage.

Davidson, E. (2006). A technological frames perspective on information technology and organizational change. *Journal of Applied Behavioral Science*, **42**(1), 23–39.

DeSanctis, G., & Monge, P. (1999). Communication processes for virtual organizations. *Organization Science*, **10**(6), 693–703.

DeSanctis, G., & Poole, M. S. (1994). Capturing the complexity in advanced technology use: Adaptive structuration theory. *Organization Science*, **5**(2), 121–147.

DuBois, P. A. (2004). *Crashworthiness engineering course notes*. Livermore, CA: Livermore Software Technology Corporation.

Edmondson, A. C., Bohmer, R. M., & Pisano, G. P. (2001). Disrupted routines: Team learning and new technology implementation in hospitals. *Administrative Science Quarterly*, **46**(4), 685–716.

Fulk, J. (1993). Social construction of communication technology. *Academy of Management Journal*, **36**(5), 921–951.

Fulk, J., Schmitz, J., & Ryu, D. (1995). Cognitive elements in the social construction of technology. *Management Communication Quarterly*, **8**(3), 259–288.

Fulk, J., Schmitz, J., & Steinfield, C. (1990). A social influence model of technology use. In J. Fulk & C. Steinfield (Eds.), *Organizations and communication technology* (pp. 117–140). Newbury Park, CA: Sage.

Fulk, J., Steinfield, C. W., Schmitz, J., & Power, G. J. (1987). A social influence model of media use in organizations. *Communication Research*, **14**(5), 529–552.

Giddens, A. (1984). *The constitution of society*. Berkeley, CA.: University of California Press.

Glaser, B. (1978). *Theoretical sensitivity*. Mill Valley, CA: Sociological Press.

Glaser, B., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine.

Goffman, E. (1983). The interaction order. *American Sociological Review*, **48**(1), 1–17.

Griffith, T. L., & Northcraft, G. B. (1996). Cognitive elements in the implementation of new technology: Can less information provide more benefits? *MIS Quarterly*, **20**(1), 99–110.

Henderson, K. (1998). *On line and on paper: Visual representations, visual culture, and computer graphics in design engineering*. Cambridge, MA: MIT Press.

Hirscheim, R. A. (1986). *Office automation: A social and organizational perspective*. New York: Wiley.

Hughes, T.J. R. (2000). *The finite element method*. Meneola, NY: Dover.

Jackson, M. H. (1996). The meaning of "communication technology": The technology-context scheme. In B. Burleson (Ed.), *Communication yearbook* (Vol. 19, pp. 229–267). Beverly Hills, CA: Sage.

Jackson, M. H., Poole, M. S., & Kuhn, T. (2002). The social construction of technology in studies of the workplace. In L. A. Lievrouw & S. Livingstone (Eds.), *Handbook of new media: Social shaping and consequences of icts* (pp. 236–253). London: Sage.

Jian, G. (2007). "Omega is a four-letter word": Toward a tension-centered model of resitance to information and communication technologies. *Communication Monographs*, **74**(4), 517–540.

Johnson, B., & Rice, R. E. (1987). *Managing organizational innovation*. New York: Columbia University Press.

Joshi, K. (1991). A model of users' perspectives on change: The case of information systems technology implementation. *MIS Quarterly*, **15**(2), 229–240.

Kraut, R. E., Rice, R. E., Cool, C., & Fish, R. S. (1998). Varieties of social influence: The role of utility and norms in the success of a new communication medium. *Organization Science*, **9**(4), 437–453.

Lapointe, L., & Rivard, S. (2005). A multivlevel model of resistance to information technology implementation. *MIS Quarterly*, **29**(3), 461–491.

Leonard-Barton, D., & Deschamps, I. (1988). Managerial influence in the implementation of new technology. *Management Science*, **34**(10), 1252–1265.

Leonardi, P. M. (2003). Problematizing "new media": Culturally based perceptions of cell phones, computers, and the Internet among United States Latinos. *Critical Studies in Media Communication*, **20**(2), 160–179.

Leonardi, P. M. (2007). Activating the informational capabilities of information technology for organizational change. *Organization Science*, **18**(5), 813–831.

Leonardi, P. M. (2008). Indeterminancy and the discourse of inevitability in international management. *Academy of Management Review*, **33**(4), 975–984.

Leonardi, P. M., & Bailey, D. E. (2008). Transformational technologies and the creation of new work practices: Making implicit knowledge explicit in task-based offshoring. *MIS Quarterly*, **32**(2), 411–436.

Leonardi, P. M., & Barley, S. R. (2008). Materiality and change: Challenges to building better theory about technology and organizing. *Information and Organization*, **18**, 159–176.

Lewis, L. K., & Seibold, D. R. (1993). Innovation modification during intraorganizational adoption. *Academy of Management Review*, **18**(2), 322–354.

Lewis, L. K., & Seibold, D. R. (1996). Communication during intraorganizational innovation adoption: Predicting users' behavioral coping responses to innovations in organizations. *Communication Monographs*, **63**(2), 131–157.

Lewis, L. K., & Seibold, D. R. (1998). Reconceptualizing organizational change implementation as a communication problem: A review of literature and research agenda. In M. E. Roloff (Ed.), *Communication yearbook* (Vol. 21, pp. 93–151). Thousand Oaks, CA: Sage.

Majchrzak, A., Rice, R. E., Malhotra, A., King, N., & Ba, S. L. (2000). Technology adaptation: The case of a computer-supported inter-organizational virtual team. *MIS Quarterly*, **24**(4), 569–600.

Manross, G., & Rice, R. E. (1986). Don't hang up: Organizational diffusion of the intelligent telephone. *Information and Management*, **10**(3), 161–175.

Marakas, G. M., & Hornik, S. (1996). Passive resistance misuse: Overt support and covert recalcitrance in IS implementation. *European Journal of Information Systems*, **5**(3), 208–220.

Markus, M. L. (1983). Power, politics, and misimplementation. *Communications of the ACM*, **26**(6), 430–444.

Markus, M. L. (2004). Technochange management: Using IT to drive organizational change. *Journal of Information Technology*, **19**, 4–20.

Martinko, M. J., Henry, J. W., & Zmud, R. W. (1996). An attributional explanation of individual resistance to the introduction of information technologies in the workplace. *Behaviour & Information Technology*, **15**(5), 313–330.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: A source book of new methods* (2nd ed.). Beverly Hills, CA: Sage.

Nass, C., & Mason, L. (1990). On the study of technology and task: A variable based approach. In J. Fulk & C. Steinfield (Eds.), *Organizations and communication technology* (pp. 46–67). Newbury Park, CA: Sage.

Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, **3**(3), 398–427.

Orlikowski, W. J. (2000). Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science*, **11**(4), 404–428.

Orlikowski, W. J. (2005). Material works: Exploring the situated entanglement of technological performativity and human agency. *Scandinavian Journal of Information Systems*, **17**(1), 183–186.

Orlikowski, W. J., & Gash, D. C. (1994). Technological frames: Making sense of information technology in organizations. *ACM Transactions on Information Systems*, **12**, 174–207.

Pinch, T. J. (1996). The social construction of technology: A review. In R. Fox (Ed.), *Technological change: Methods and themes in the history of technology* (pp. 17–35). Amsterdam: Harwood.

Pinch, T. J., & Bijker, W. E. (1984). The social construction of facts and artifacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science*, **14**, 399–441.

Poole, M. S., & DeSanctis, G. (1990). Understanding the use of group decision support systems: The theory of adaptive structuration. In J. Fulk & C. Steinfield (Eds.), *Organizations and communication technology* (pp. 173–193). Newbury Park, CA: Sage.

Poole, M. S., & DeSanctis, G. (1992). Microlevel structuration in computer-supported group decision making. *Human Communication Research*, **19**(1), 5–49.

Rice, R. E. (1987). Computer-mediated communication and organizational innovation. *Journal of Communication*, **37**(4), 65–94.

Rice, R. E. (1991). Organic organizations and centralized units: Use, contexts, and outcomes of word processing. In J. Morell & M. Fleischer (Eds.), *Advances in the implementation and impact of computer systems* (Vol. 1, pp. 111–141). New York: JAI Press.

Rice, R. E., & Aydin, C. (1991). Attitudes toward new organizational technology: Network proximity as a mechanism for social information processing. *Administrative Science Quarterly*, **36**(2), 219–244.

Rice, R. E., Collins-Jarvis, L., & Zydney-Walker, S. (1999). Individual and structural influences on information technology helping relationships. *Journal of Applied Communication Research*, **27**(4), 285–309.

Rice, R. E., & Gattiker, U. (2001). New media and organizational structuring. In F. M. Jablin & L. L. Putnam (Eds.), *The new handbook of organizational communication: Advances in theory, research, and methods* (pp. 544–581). Thousand Oaks, CA: Sage.

Rice, R. E., & Rogers, E. M. (1980). Reinvention in the innovation process. *Knowledge*, **1**, 499–514.

Rice, R. E., & Shook, D. (1990). Relationships of job categories and organizational levels to use of communication channels, including electronic mail: A meta-analysis and extension. *Journal of Management Studies*, **27**, 195–229.

Robey, D., & Boudreau, M.-C. (1999). Accounting for the contradictory organizational consequences of information technology: Theoretical directions and methodological implications. *Information Systems Research*, **10**(2), 167–185.

Robey, D., & Sahay, S. (1996). Transforming work through information technology: A comparative case study of geographic information systems in county government. *Information Systems Research*, **7**(1), 93–110.

Rose, J., Jones, M. R., & Truex, D. (2005). Socio-theoretic accounts of IS: The problem of agency. *Scandinavian Journal of Information Systems*, **17**(1), 133–152.

Strauss, A. (1978). *Negotiations*. San Francisco: Jossey-Bass.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Thousand Oaks, CA: Sage.

Suchman, L. (2000). Organizing alignment: A case of bridge-building. *Organization*, **7**(2), 311–327.

Vaast, E., & Walsham, G. (2005). Representations and actions: The transformation of work practices with IT use. *Information and Organization*, **15**, 65–89.

Vincenti, W. G. (1990). *What engineers know and how they know it: Analytical studies from aeronautical history*. Baltimore: Johns Hopkins University Press.

人们为什么拒绝新技术并阻碍他们偏好的组织变化？
探究社会互动和物质性之间的不一致

Paul M. Leonardi
西北大学

摘要

本论文探究用户对新技术的解释和组织变化失败两者之间的关系。我认为人们对新技术的解释不仅基于他们和其他人的讨论，而且基于他们对技术之物质性特征的直接使用上。通过对一家大型汽车工程公司执行和使用计算机模仿技术的人种数据进行定性和定量分析，我发现工程师与经理、同事及顾客之间的交流使他们形成了对有关技术能力的解释；与此同时，与互补性技术的物质性特征进行的互动使工程师得出另一种解释，即新的模仿技术并非实现上述特定目标的有效工具。我展示了社会互动的解释对人们在将来使用该技术是受人们物质性互动解释的影响的。 我然后证明在这个特定情境中，工程师通过减少新技术的使用而无意中阻碍了他们非常热衷的组织变化。我讨论了用户和他人互动所取信息和技术物质性特征的互动所取信息之间的不一致怎样导致计划组织变化的失败。

Warum lehnen Menschen neue Technologien ab und verhindern so Veränderungen im Unternehmen, die sich eigentlich befürworten? Eine Untersuchung zur Schieflage zwischen sozialer Interaktion und Materialität

Dieser Artikel untersucht das Verhältnis zwischen der Deutung neuer Technologien durch die Nutzer und dem Scheitern von Veränderungen auf Unternehmensebene. Ich argumentiere, dass Menschen ihre Bewertung von neuen Technologien nicht nur basierend auf ihren Gesprächen mit anderen anstellen, sondern auch auf direkten Weg durch die Nutzung der stofflichen Eigenschaften dieser Technologien. Anhand einer qualitativen und quantitativen Analyse von ethnographischen Daten zur Einführung und Nutzung von Computersimulationstechnologie in einer großen Automobilfirma, verdeutliche ich, dass die Kommunikation der Ingenieure mit den Führungskräften, Kollegen und Kunden dazu führte, dass sie eine Vorstellung darüber entwickelten, was die neue Technologie leisten sollte, während ihre Interaktion mit den materiellen Eigenschaften anderer Technologien dazu führte, dass sie die neue Simulationstechnologie eben nicht als ein effizienten Werkzeug für den spezifischen Gebrauch sahen. Ich zeige auf, wie diese Einschätzung, die sich aus der materiellen Interaktion entwickelt hat, vom Einfluss der Einschätzungen, die durch soziale Interaktion entstanden sind, moderiert wird und die Bereitschaft, diese Technologie in der Zukunft einzusetzen, bestimmt. Ich zeige weiterhin, dass die Ingenieure in diesem speziellen Fall Veränderungen auf Unternehmensebene unbeabsichtigt verhindern, indem sie die neue Technologie weniger nutzen, obwohl sie diese eigentlich befürworten. Zum Schluss diskutiere ich, wie diese Schieflagen zwischen der Information, die durch die Interaktion der Nutzer mit anderen zustande kommt und die Information aus den materiellen Eigenschaften der Technologie, zum Scheitern geplanter Unternehmensentwicklungen führen kann.

Pourquoi les gens rejettent-ils les nouvelles technologies et pourquoi bloquent-ils des changements organisationnels qu'ils approuvent? Une exploration des décalages entre les interactions sociales et la matérialité

Résumé

Cet article explore le lien entre les interprétations qu'ont les usagers d'une nouvelle technologie et l'échec d'un changement organisationnel. Je soutiens que les gens forment des interprétations d'une nouvelle technologie non seulement à partir de leurs conversations avec d'autres personnes, mais aussi à travers leur usage direct des éléments matériels de la technologie. L'étude s'appuie sur l'analyse qualitative et quantitative de données ethnographiques recueillies à propos de l'implantation et de l'utilisation d'une technologie de simulation par ordinateur dans une grande entreprise d'ingénierie automobile. Je démontre que la communication des ingénieurs avec les gestionnaires, les collègues et les clients les a menés à développer une certaine interprétation de ce que la technologie était censée faire, alors que leurs interactions avec les éléments matériels des technologies complémentaires les a menés à développer une interprétation selon laquelle la nouvelle technologie de simulation n'était pas un bon outil pour cet objectif particulier. Je démontre la façon par laquelle les interprétations développées à partir des interactions matérielles des gens réduisent les effets des interprétations développées dans les interactions sociales sur l'empressement à utiliser la technologie à l'avenir. Je montre ensuite que, dans cette situation spécifique, les ingénieurs ont par inadvertance bloqué un changement organisationnel qu'ils approuvaient pourtant fortement, en réduisant leur utilisation de la nouvelle technologie. Je conclus par une discussion de la manière dont les décalages entre l'information générée dans les interactions des usagers avec d'autres personnes et avec les

éléments matériels des technologies peuvent mener à l'échec de changements organisationnels planifiés.

Why Do People Reject New Technologies and Stymie Organizational Changes of Which They're in Favor? Exploring Misalignments Between Social Interactions and Materiality

왜 사람들은 새로운 기술을 거부하고 그들이 선호하는 조직적 변화를 방해하는가?

사회적 상호작용들과 물직화 사이의 잘못된관계에 대한 연구

Paul M. Leonardi
Department of Communication Studies
Department of Industrial Engineering and Management Sciences
Northwestern University

요약

본 논문은 사용자들이 새로운 기술들을 어떻게 해석하는가와 조직적 변화의 실패사이의 관계를 연구한 것이다. 본 연구는 사람들은 새로운 기술에 대한 해석에 있어 이를 타인과의 대화에 기초해서뿐만 아니라 기술 내용들의 직접적인 사용을 통해 형성한다는 것을 보여주고 있다. 주요 자동차 엔진 회사에서의 컴퓨터 응용 기술의 응용과 사용에 대한 인류학적 자료의 질적, 그리고 양적 분석을 통해, 본 연구는 엔지니어가 관리자, 동료, 그리고 고객들과 하는 커뮤니케이션은 그들로 하여금 기술이 무엇인가에 대한 해석을 발전시킨다는 것을 보여주고 있다. 반면, 보완적인 기술들의 물질적 내용들과의 상호작용은 새로운 응용기술들이 특정한 목적을 위해 유용한 도구가 아니라는 개념을 이끈다는 것을 보여주고 있다. 본 논문은 사람들의 물질적 상호작용으로부터 발전된 해석들이 미래에 기술을 사용하려는 사회적 의도를 통해 발전된 해석들의 영향을 어떻게

중재하는가를 설명하고 있다. 그리고 나서, 본 논문은 이러한 특별한 상황에서, 기술자들은 새로운 기술의 사용을 줄이는 것에 의해 그들이 선호했던 조직의 변화를 의도하지 않게 방해했다는 것을 증명하고 있다. 본 논문은 사용자가 다른 사람들과의 상호작용과 기술 물질형태와의 상호작용을 통해 생산된 정보사이의 불일치가 계획된 조직적 변화를 이끌지 못하는 것을 논의하였다.

¿Por Qué la Gente Rechaza las Nuevas Tecnologías y Bloquea los Cambios Organizacionales que Ellos Acuerdan?

Explorando los Desalineamientos entre las Interacciones Sociales y la Materialidad

Resumen

Este artículo explora la relación entre las interpretaciones de los usuarios de nuevas tecnologías y los fracasos del cambio organizacional. Sugiero que la gente forma interpretaciones sobre la nueva tecnología basada no solo en las conversaciones con otros, sino también a través del uso de las características materiales de la tecnología. A través de análisis cuantitativos y cualitativos de datos etnográficos sobre la implementación y uso de una simulación computarizada de tecnología en una firma de ingeniería automotriz, muestro que la comunicación de los ingenieros con los gerentes, compañeros de trabajo, y clientes los llevó a desarrollar una interpretación sobre lo que la tecnología supuestamente hace mientras que sus interacciones con las características materiales de las tecnologías complementarias los llevó a desarrollar una interpretación que la nueva simulación tecnológica no era una herramienta eficiente para ese propósito especifico. Muestro cómo las interpretaciones desarrolladas de las interacciones materiales de la gente moderaron los efectos de las interpretaciones desarrolladas a través de las interacciones sociales de buena voluntad en el uso de la tecnología en el futuro. Demuestro que, en este contexto particular, los ingenieros bloquearon inadvertidamente el cambio organizacional que ellos favorecían mediante la reducción del uso de la nueva tecnología. Concluyo con una discusión sobre cómo el desalineamiento entre la información generada en las interacciones de los de los usuarios con otros y con las

características materiales de la tecnología puede llevar al fracaso del cambio organizacional planeado.