WHEN DOES TECHNOLOGY USE ENABLE NETWORK CHANGE IN ORGANIZATIONS? A COMPARATIVE STUDY OF FEATURE USE AND SHARED AFFORDANCES

Paul M. Leonardi
Department of Communication Studies and Department of Industrial Engineering and Management Sciences, Northwestern University, 2240 Campus Drive, Evanston, IL 60208 U.S.A. {leonardi@northwestern.edu}

The goal of this study is to augment explanations of how newly implemented technologies enable network change within organizations with an understanding of when such change is likely to happen. Drawing on the emerging literature on technology affordances, the paper suggests that informal network change within interdependent organizational groups is unlikely to occur until users converge on a shared appropriation of the new technology’s features such that the affordances the technology enables are jointly realized. In making the argument for the importance of shared affordances, this paper suggests that group-level network change has its most profound implications at the organization level when individuals use the same subset of a new information technology’s features. To explore this tentative theory, we turn to a comparative, multimethod, longitudinal study of computer-based simulation technology use in automotive engineering. The findings of this explanatory case study show that engineers used the new technology for more than three months, during which time neither group experienced changes to their advice networks. Initially, divergent uses of the technology’s features by engineers in both groups precluded them from being able to coordinate their work in ways that allowed them to structure their advice networks differently. Eventually, engineers in only one of the two groups converged on the use of a common set of the technology’s features to enact a shared affordance. This convergence was necessary to turn the technology into a resource that could collectively afford group members the ability to compare their simulation outputs with one another and, in so doing, alter the content and structure of the group’s advice network. The implications of these findings for the literatures on technology feature use, affordances, social networks, and post-adoption behaviors in organizations are discussed.

Keywords: Technology implementation, organizational change, advice networks, feature use, affordances, frames

Introduction

It is no secret that employees get their work done by seeking the advice of others. Numerous field studies—from Blau’s (1955) classic study of federal business auditors to Perlow’s (1997) examination of software engineers—have shown that most people’s work could not be accomplished if they did not regularly turn to colleagues for advice. Indeed, informal advice networks (networks of communication in which ties represent one person seeking advice about work related issues from another) within organizations are consistently shown to be important engines of productivity and social support.
because they enable the movement of pertinent information among employees (Cross et al. 2001; Gibbons 2004). For this reason, it is not surprising that many researchers have explored how (Fulk 1993; Sykes et al. 2008) and when (Kane and Alavi 2008; Rice and Aydin 1991) informal advice networks affect the adoption and use of new information technologies. It is surprising, however, that when considering the ways that newly implemented information technologies might be implicated in the shifting dynamics of advice networks, research has been highly skewed toward a focus on how changes occur and has often overlooked when such changes are likely.

For example, early work by Barley (1990) and Burkhart and Brass (1990) offered complementary theories about how new information technologies changed advice networks in organizations. Both studies showed that users faced steep learning curves when attempting to operate the new technology. Those individuals who developed operational expertise quickly were sought by their colleagues for advice about how to use the technology and gained both prestige and power amongst their peers. Building on this early work, Shultze and Orlowski (2004) showed how such changes in informal consultation patterns were wrought slowly and in response to formal alterations to work roles that managers made to take advantage of a new technology’s capabilities and shortcomings. Leonardi (2007) showed how the decision to use features of a technology that were not used previously gave employees access to new information about other people’s expertise and this new information led them to consult people they had not consulted before. Yet although each of these studies, and others like them (e.g., Black et al. 2004; Constant et al. 1996), develop useful theories about how (the mechanisms by which) new technologies change advice networks, they are largely silent on when (under what conditions) such change is likely to happen.

This issue of when a newly implemented technology will enable changes in an organization’s informal advice networks within workgroups or departments is both theoretically and practically important. If people are able to do their work through use of their established networks, and those networks change, they may be no longer able to complete tasks with equal quality or speed. Or, if people’s networks provide them with poor or ineffectual information, a change in communication partners or topics of conversation may lead to increases in work productivity and job satisfaction. Obviously, such group-level dynamics can have important consequences for the organization as a whole. But there is no guarantee that a new technology, even one that enables access to radically new information, will lead to changes in informal advice networks. In fact, the evidence suggests that even in situations where people continue using the technology after implementation (i.e., they have not rejected it), network change does not always occur (Markus 2004; Poole and DeSanctis 2004). Unfortunately, there are few empirical studies comparing the implementation and use of identical technologies in multiple contexts. Most comparative studies focus on whether a new technology changes formal organizational positions or roles (e.g., Edmondson et al. 2001; Robey and Sahay 1996) and, consequently, overlook the informal network changes that emerge once use begins. Those comparative studies that do focus on changes in informal network dynamics tend to gloss over a discussion of when such changes happened in favor of detailed theoretical expositions about how those changes unfolded (e.g., Barley 1990; Zack and McKenney 1995).

I propose that to answer the important question of when information technologies will bring changes to organizational advice networks requires a focus on the way members of a social group actually use the features of the technology. IS scholars have argued that advanced information technologies are not monolithic; rather, they are comprised of many features that can be used—often independently of one another—in a variety of ways (Burton-Jones and Straub 2006; Orlowski and Iacono 2001). Researchers who have taken to multilevel theorizing have suggested that the various individuals who make up a social group may choose to use different features of the technology than those chosen by their colleagues, and that these differential uses can have important implications for outcomes at the organization level of analysis (Burton-Jones and Gallivan 2007; Kane and Labianca 2011).

Using these insights as a starting point, I turn to the emerging literature on technology affordances to suggest that informal network change within organizations whose groups engage in interdependent tasks is unlikely to occur until users converge on a shared appropriation of the new technology’s features such that the affordances the technology provides are jointly realized. In making the argument for the importance of shared affordances, this paper suggests that group-level network change has its most profound implications at the organization level when individuals use the same subset of a new information technology’s features. In what follows, I build the theoretical rationale for this claim and explore its empirical tenability through a comparative, multimethod, longitudinal study of computer-based simulation technology use in automotive engineering.

Feature Use and Shared Affordances

In the past several decades, a good deal of evidence has accumulated by constructivist researchers to debunk strong
As Markus and Silver (2008) observe, the focus on the feature level of the "feature set" (1994, p. 126), instead of at the level of the "feature set" (1994, p. 126), instead of at the level of the artifact, writ large, they were able to show considerable variation in what features people used when they used them. The early findings from this line of research demonstrated that groups that used the technology as the designers intended had better consensus in group decision making than groups that used it in unexpected ways (Poole and DeSanctis 1992; Poole et al. 1991). At the heart of their argument was the assumption that effective group decision making followed particular principles and that the features included in the group decision support system were there specifically (DeSanctis and Poole designed the system they studied) to afford users the ability to realize those principles (e.g., anonymity of response, turn taking, etc.). Thus, they showed that groups whose members used the technology's features in similar ways experienced better outcomes than groups whose members did not; but, they argued, there was no way to predict what pattern of use a group would end up with (Poole and DeSanctis 2004).

Although it is this latter insight about emergence that most often finds its way into citations to AST, DeSanctis and Poole's empirical findings about feature use affording particular outcomes also have important theoretical consequences. As Markus and Silver (2008) observe, the focus on the feature set comprising the technology, and how users take advantage of it, is essential for understanding what it is about IT that may contribute to the behavioral and social outcomes of IT use, when such effects occur. This is not to say that technology is the only, or even the most important, contributor to IT effects, but merely that it may matter (p. 610).

In other words, it is the capabilities of the technology, just as much as the choices people make about how to use those capabilities, which explain the ultimate effects that technologies have on social structures. They are two sides of the same coin.

To further elaborate this relationship between the technology's features and the way people use those features, Markus and Silver turned to a discussion of affordances. The concept of affordances has a long history in studies of ecological psychology (Chemero 2003; Gibson 1986). The idea is that objects have properties (or features, in the context of IT use) and animals that make use of objects have their own physical characteristics and a host of needs. Any animal can perceive an object's features (e.g., its flatness or roundness), but the utility of those features—what they afford (e.g., walking on or rolling on)—is "relative to the posture and behavior of the animal being considered" (Gibson 1986, pp. 127-128). In this way, affordances are relational (Hutchby 2001; Zammuto et al. 2007). They are given neither by the technology nor by the perceiver alone. Information systems researchers who adopt a relational view of affordances stress that people's goals shape what they come to view the features of the technology as affording them the ability to do (Leonardi 2011b; Markus and Silver 2008). Yet considering the relationship between affordances, features, and effects suggests a double bind of sorts in that users will appropriate certain features of a technology only when they perceive that those features offer them affordances for action, but if those features are not appropriated, their material qualities cannot afford social structural change. For this reason, Markus and Silver argue that affordances should be "understood as potentially necessary (but not necessary and sufficient) conditions for 'appropriation moves' (IT Uses) and the consequences of IT use" (p. 625).

A growing number of studies have shown that one technology can support multiple affordances, and, consequently, that each member of the same social group can enact a different affordance or set of affordances when using the same technology (Davern et al. 2012; Kaptelinin and Nardi 2006). One reason that a multiplicity of affordances can be enacted from the use of one technology is due to the fact that affordances do not
exist absent a user’s intentions or goals (Markus and Silver 2008). If people hold different goals, they may enact different affordances from the same technology; that is, they will use its features in ways that are distinct from how other group members use them. But different goals are not the only reason that the enactment of diverse affordances is possible within a social group. Even when using the exact same technology, users of a defined social group can differentially appropriate the features comprising the technology’s features set. DeSanctis and Poole referred to this reality as the “repeating decomposition problem” (p. 124). With multiple members in a group, and multiple features available for use, the possible number of affordances that may be enacted when different individuals use the technology is very large.

For this reason, I suggest distinctions between the concepts of individualized affordance, collective affordance, and shared affordance. An individualized affordance is an affordance that someone enacts when using a technology’s features, but that affordance is not common to his or her workgroup or department. An individualized affordance will benefit the person who enacted it, but that affordance may not be available to everyone else in the group. Consequently, the enactor of that affordance might be able to use the technology to do something that others cannot. A person who enacts an individualized affordance may gain power or status within a group or become more central in group discussions and influence work by virtue of his or her ability to do things others can not (Kane and Borgatti 2011). Consequently, an individualized affordance is an individual-level construct.

A collective affordance is an affordance that is collectively created by members of a group, in the aggregate, which allows the group to do something that it could not otherwise accomplish. A collective affordance may be the result of pooled individualized affordances. Collective affordances are likely to arise when work is highly specialized (individuals conduct different tasks) and where the interdependence is either nonexistent or pooled: individuals work on their own tasks and those tasks are aggregated to produce a final output (Thompson 1967). In situations of pooled interdependence, group members do not have to use the technology in the same way because varied patterns of use afford different capabilities that may be useful for the different kinds of work group members carry out. In the enactment of a collective affordance, there is, as Oborn et al. (2011, p. 563) describe, “unity in diversity”: diverse use of the same system can afford different users distinct capabilities that are all essential for allowing the group, as a whole, to complete its work. As such, a collective affordance is a group-level construct.

A shared affordance is an affordance that is shared by all members of a group. A shared affordance is distinct from a collective affordance because the former represents similar use of the technology’s features by all members while the latter represents differential feature use that is necessary for completing noninterdependent tasks that, when pooled, achieve the group-level goal. Whereas collective affordances are likely to arise in teams characterized by pooled interdependence, shared affordances are perhaps more common in teams whose work is characterized by high degrees of reciprocal interdependence. In situations characterized by reciprocal interdependence “group members must interact and depend on each other in order for the group to accomplish its work” (Guzzo and Shea 1992, p. 296). Reciprocally interdependent group members perform similar tasks in a project and work closely together to make sure that their outputs are well coordinated. In work of this type, individuals who share similar patterns of technology use are afforded the same capabilities and, consequently, can continue to coordinate their work. If workgroups characterized by a high degree of reciprocal interdependence have members who enact varied individualized affordances from use of different features (which may or may not aggregate to a collective affordance at the group level), they may not be able to complete their work successfully and performance, at both the individual level and the group level, will suffer. But if those individuals in the group enact a shared affordance such that the entire group now has a new resource with which to work, they can easily coordinate their work and achieve individual and group goals.

Proponents of multilevel perspectives on information systems use argue convincingly that at the group level, regardless of the type of tasks conducted, use of a new technology can follow either a shared or a configurational structure (Burton-Jones and Gallivan 2007; Kane and Labianca 2011; Kozlowski and Klein 2000). A shared structure of use means that everyone in the group uses the technology at roughly the same frequency and they employ its features in roughly the same way. For example, if a technology were comprised of 10 features, a shared usage structure would mean that everyone uses, say, features 1, 5, 7, and 9. By contrast, a configurational structure of use means that people in a group might be using the technology at roughly the same frequency, but they are using its features in various configurations. For example, Person A might use a configuration of features 1, 3, and 5, Person B might use a configuration of features 2, 4, and 6, and Person C might use a configuration of features 7, 8, and 9. To further complicate matters, Person A might use his configuration of features for task Alpha while Person B uses her configuration for task Beta (for extended discussion of various configurational possibilities, see Burton-Jones and Gallivan 2007, pp. 668-669). Groups that converge upon a shared structure of feature use are likely to enact a shared affordance while
groups whose members diverge from one another in their use of features to constitute a configurational structure are likely to enact collective affordances.

In groups with a high degree of reciprocal task interdependence, which will be the focus of this study, enacting a shared affordance (as opposed to an individualized or collective affordance) through a shared structure of feature use at the group level may be necessary for network change that is meaningful at the organization level. Building on threshold models of network dynamics (e.g., Granovetter 1978), Monge and Contractor (2001) suggested that inclusiveness is an important predictor of whether communication networks change within organizations whose work is characterized by high degrees of task interdependence. They argue that in any group there is a threshold for the number of people in a communication network that need to act in similar ways to determine “whether the group as a whole can achieve the critical mass necessary for rapid and widespread collective action” (p. 457). In other words, network change that has effects at the organization level of analysis (e.g., change to what the organization does or how it does it) requires a significant number of people in a reciprocally interdependent social group to change their behaviors in similar ways. As Monge and Contractor (2003) later observed, collective action typically brings about network changes with significantly different organizational implications than individualized action. For example, if everyone in a group suddenly had access to new information, members of the group would likely together (collective action) shift the structure of their communication partners (so as to talk with people who are now more relevant) and the content flowing along the ties would also change (e.g., instead of talking about topic A, people now talk about topic B). Such changes in structure and content can dramatically alter the way the organization, as a whole, functions or how it conceives of its identity (Golden-Biddle and Rao 1997). But if only one person has access to information that others do not, group members might shift the structure of their ties or the content of their communication with that one person, but not with everyone else in the network (individualized action). Thus, while in both cases network change would have occurred, the changes enabled by collective action would affect the organization, writ large, because they cascade out beyond dyadic linkages to the entire network, while the changes enabled by individualized action would more likely be more dyadic in nature and, consequently, will not tend to reverberate up levels of analysis to make meaningful change at the organization level (Rowley 1997).

By giving people similar resources to use in their work, the enactment of shared affordances among members of reciprocally interdependent groups may provide the collective action necessary to instigate the kinds of changes to informal advice networks that have meaningful impacts for the organization. Because information is what flows along ties in advice networks, particular appropriations of a technology’s features that provide new information to all group members may be the catalyst for change in the structure and content of the network (Leonardi 2007; Nan 2011). Once group members appropriate a technology’s features in ways that provide everyone with new information, the information will afford the group, as a whole, the ability to change who they talk to and, by association, how they work.

In weaving together the discussions of feature use, shared affordances, and changes in advice networks, I have developed a tentative theory of when use of a new technology will lead to network changes that have important organizational repercussions: Once a group of users begins to appropriate the features of the technology in shared ways. The theory suggests that the reason convergence in feature use among members is necessary for advice network change is that the use of certain features affords people capabilities to work in new ways. Convergence toward a shared structure of feature use allows group members to enact a shared affordance, which, in the context of information technologies and advice networks, means that everyone has access to new information such that group members are together afforded the possibility to change how they interact with one another. To explore the empirical tenability of this theory, I report the findings from a comparative analysis of two automotive engineering teams using a newly implemented simulation technology for crashworthiness analysis.

Background and Methods

Crashworthiness Engineering and CrashLab

Autoworks (a pseudonym, as are all names used in this paper) is a large U.S. automobile manufacturer. I focused my data collection efforts on the work of engineers in Autoworks’ Safety and Crashworthiness Engineering Division (hereafter, Safety). Crashworthiness engineers evaluate the performance of a vehicle’s structure by assessing its ability to protect occupants during an impact. Because the cost of administering and recording the data for physical crash tests is often prohibitive, Autoworks requires its engineers to use computer-based simulation technologies to make recommendations for early-stage vehicle design. In an attempt to reduce the time and effort it took engineers to set-up and analyze simulations, as well as to standardize the assumptions engineers used while doing so, innovators in Autoworks’ research and development
division developed a new technology called CrashLab. CrashLab was a software application developed to automate many of the preprocessing (preparing a simulation model to be submitted to a solver) and post-processing (analyzing the results of the solved model) tasks that engineers had, up to that point, to perform manually with existing simulation tools.

By implementing a technology that automated simulation work, managers intended to bring about a significant organizational change. The vast majority of engineering organizations make a clear distinction 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 (Vincenti 1990). In fact, the boundaries that demarcate most formal roles and job responsibilities in engineering organizations (for example, between design engineers and analysis engineers) 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 time spent performing routine model building activities. Shifting the focus of effort from model building to analysis activities marks an important, significant, and highly desirable organizational change in most engineering firms (Collins 2003). Autoworks’ managers wished to see such changes in their engineering functions. Instead of engineers spending so much time setting up simulation models, they hoped CrashLab would help engineers move to analysis more quickly and come up with more innovative vehicle design solutions. In September 2005 they implemented CrashLab to instigate such change.

Data Collection

Between July 2004 and August 2006, I spent nine months engaged in ethnographic data collection at Autoworks. I gained access to Autoworks through an internship program allowing researchers to work in the company’s R&D division on projects related to understanding engineering and technical work. The nine months of observation time were divided into four periods of residency in Safety. The first period was July–August 2004, one year before CrashLab was implemented. The second period was August–November 2005. I began observations three weeks before CrashLab was officially launched and I remained in Safety for 13 more weeks to observe engineers make their initial interpretations and uses of the new technology. I returned for a third period, March–April 2006, which began 26 weeks after CrashLab was implemented. Finally, I returned to Autoworks for a third period, July–August 2006, staying through the 50th week after CrashLab was implemented.

I conducted field observations about three related activities: the work of crashworthiness engineers before CrashLab was implemented, the activities of developers, trainers, and managers during the implementation process, and the work of engineers after CrashLab was implemented. During the periods in which I was a resident in Safety I utilized three primary data sources: observations made of informants at work, sociometric surveys distributed to informants, and logs kept by informants tracking their use of CrashLab. I outline each of these data collection procedures below.

Observations

Safety was comprised of a number of vehicle program groups. All vehicles that shared a common chassis, powertrain, and other components were designed on a common platform. Each platform supported various vehicle programs—what ordinary drivers refer to as “models.” During Period 1, I conducted observations with engineers in various vehicle program groups. The purpose of these early observations was to generate a detailed understanding of the way that engineers worked before the implementation of CrashLab, so as to have a basis from which to compare whether or not their work changed after implementation of the new technology occurred. In this way, I followed guidelines for explanatory case research in information systems and the social sciences (Dubé and Paré 2003; George and Bennett 2005), which suggest employing a longitudinal design so as to be able to explain the genesis of change, rather than infer it, by tracing causal processes from one time period to their effects in the next.

In Period 2, I observed 24 different events (e.g., staff meetings, formal training sessions, etc.) in which developers and managers told engineers about CrashLab and taught them how to use it. In this period, I focused my attention on two different vehicle program groups within Safety that were using CrashLab. I call these groups the Piston Group and the Strut Group. Selection of the Piston Group and the Strut groups was influenced, in part, by guidelines for explanatory case research in information systems, which suggest finding a situation in which natural controls exist such that the analyst

For a detailed discussion of how and why CrashLab was developed, see Leonardi (2011a).
can structurally eliminate from consideration variables that could have affected the outcome in question (Dubé and Paré 2003, p. 606). The Piston group, comprised of 19 engineers, and the Strut group, comprised of 17 engineers, were chosen because they were very similar to one another. Engineers in both groups were trained in similar academic disciplines, they had all worked on a variety of program groups in Safety before joining their current group, and they conducted identical analyses on different vehicles. The average age of informants in the Strut Group was 34 years old with an average tenure at Autoworks of 7 years. The average age of informants in the Piston Group was 36 years old with an average tenure of 7.5 years at the company. Ethnic and gender breakdowns were virtually identical for the two groups. The two groups sat in different parts of a large building and their work was in no way interdependent. There were no changes in personnel in either of these groups throughout the course of this study. Managers in both of the groups were equally as excited by the prospect of using CrashLab to move engineering work from modeling to analysis and both managers equally encouraged their engineers to use the new technology. Every engineer in the Piston and Strut groups had the identical suite of tools installed on his/her desktop because they conducted identical work. In my observations before the implementation of CrashLab, no differences were noted in the frequency of particular tool used across the two groups. In short, the groups in this matched sample (Jick 1979) shared enough similarities that alternative hypotheses for network change, including hypotheses related to the current ecology of technology use, different backgrounds, varying levels of managerial support for the technology, or demographic attributes could be ruled out (Benbasat et al. 1987).

The Piston and Strut groups were chosen for two additional reasons. First, members of each group worked with each other very closely under an explicit structure of reciprocal interdependence. Within each group, several engineers worked together on a particular kind of analysis (e.g., frontal impact tests, side impact tests, or occupant injury tests) and engineers in each of these areas had to constantly coordinate their work with others because changes to one area of the vehicle (e.g., the front rails) affected analyses in other parts of the vehicle (e.g., the occupant compartment). Consequently, it is people conducting work of this type who may most need to enact shared affordances to instigate network change.

Second, both groups occupied similar temporal positions in the vehicle development process. Engineers in Safety began their most intense period of work on a vehicle program after the design had passed through a stage-gate known as Vehicle Program Initiation (VPI). Because CrashLab was a technology that would allow engineers to build and analyze simulation models, managers and developers reasoned that it would affect the most change in engineers’ work when a particular program was still flexible enough to be redesigned and reconfigured with the results from the crashworthiness analyses. Typically, this period in which engineers could still radically affect the vehicle’s design lasted from the VPI stage-gate until the Structure Vehicle Engineering Release (SVER) stage-gate, which normally occurred one year later. The Piston Group and the Strut Group completed the VPI stage-gate shortly before CrashLab was implemented. In total, I conducted 64 observations with engineers in the Piston Group and 60 observations with engineers in the Strut Group after CrashLab was implemented. Each of these observations were focused on shadowing a focal informant during their normal work for three to six hours and recording a running narrative of events, including all tasks they conducted, technologies they used, and people with whom they talked. All engineers on each group were observed on at least three separate occasions. The distribution of these observations across my periods of residency at Autoworks can be found in Table 1.

Sociometric Surveys

Sociometric surveys are instruments used to collect data on the patterns of social relations between actors. During observations before CrashLab was implemented (Period 1), I watched engineers consult their colleagues regularly about model set-up and model analysis activities. Regardless of vehicle program, all model set-up required knowledge of at least three activities: (1) how to position a barrier, (2) where to place accelerometers, and (3) how to define sections. Likewise, to execute a useful analysis, all engineers had to make decisions based on at least three key factors to improve crashworthiness performance: (1) how to change the materials used to build parts, (2) how to change the geometry of

---

3The Piston and Strut groups were chosen as sites not only because they allowed for a synchronic analysis of the same technology in use in multiple settings, but also because they allowed me to examine, in detail, how each of the groups made sense of the CrashLab over time. The diachronic analysis of the Piston Group’s reactions to CrashLab is reported in Leonardi (2009) and the Strut Group’s reaction is reported in Leonardi (2011b).

4An accelerometer is an electromechanical device used to measure forces in a vehicle impact. In physical tests, accelerometers are placed at various portions on the vehicle to measure forces. In a simulation, nodes are selected as accelerometers at which the solver will determine the accelerative forces in the model.

5To define a section is to place a virtual plane through a vehicle’s structural member. That plane is used to measure the force in that particular member.
Table 1. Summary of Observational and Interview Data Collected at Autoworks

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>With Engineers Before CrashLab (across all of Safety)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Observations</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td><strong>Implementation Activities (e.g. training sessions, staff meetings)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Observations</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td><strong>With Engineers After CrashLab (Piston Group)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>Observations</td>
<td>34</td>
<td>18</td>
<td>12</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td><strong>With Engineers After CrashLab (Strut Group)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>Observations</td>
<td>32</td>
<td>17</td>
<td>11</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td><strong>Interviews With Other Relevant Actors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Managers</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>Developers</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Vice President</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

parts, and (3) how to change the location of parts. Together, these six areas of consultation sketched an outline of the communicative exchanges constitutive of the advice networks in Safety.

To determine whether these patterns of consultation, and thus the informal advice networks changed over the course of this study, all engineers in the Piston and Strut groups were given identical sociometric surveys asking them to indicate all other engineers from whom they sought advice on the six issues (three model building and three model analysis issues) listed above. Each question, “From whom do you routinely seek advice about [insert issue]?” was followed by a roster containing the names of every engineer in Safety who worked with computer simulations (83 engineers total). The names on the roster were taken from the list of employee names on Autoworks’ most up-to-date organization chart. Respondents were asked to select as many names from the roster for each of the six areas of consultation as they felt was appropriate (i.e., the names selected were people from whom they routinely sought advice). By limiting the potential universe of engineers an informant could select (they could only select people within Safety), I was able to accrue some key advantages of a roster-based selection, namely that informants are more accurate in selecting alters (because the names are present they do not forget) and that there is often an increased reliability of results (Marsden 2005). This same survey was administered in three waves. The first wave was conducted in August 2005, just before CrashLab was implemented. The second wave was conducted three months after implementation. The only difference in the sociometric survey during the second wave was that respondents were asked to select people they routinely sought advice from during the previous three months only. The final wave was conducted in August 2006, and was again identical to the previous two waves with the one exception that respondents were asked to select people from whom they routinely sought advice in the past nine months. Across all three waves of the survey, there were only two changes in employment across safety: two people retired. These two names were removed from the final survey. After reminder emails and some direct visits to their desks, I obtained a 100 percent response rate from the 19 engineers in the Piston Group and the 17 engineers in the Strut Group across all three waves.

**Tracking Logs**

Tracking logs were the final method employed to collect data on CrashLab use at Autoworks. Each Friday, a tracking log, which contained a list of all of the major features available for use in CrashLab (see Table 2), was sent to the informants in the Piston and Strut groups via e-mail. The engineers were asked to select which features, if any, they used during that week. A response rate of 72 percent was achieved for tracking logs over the 50 weeks they were sent to engineers in both groups.

**Data Analysis**

The analysis of data proceeded in three stages. In the first stage, the completed tracking logs were used to uncover which of CrashLab’s 12 features informants in the Piston and
**Table 2. Description of CrashLab’s 12 Major Features**

<table>
<thead>
<tr>
<th>Material Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Check Model</td>
<td>Reviews assembled model to assure there are no missing parts, no part penetrations, and that material properties have been assigned to all parts.</td>
</tr>
<tr>
<td>2. Create Sections</td>
<td>Guides user through the process of making section cuts at predefined areas of a model so as to measure accelerative forces produced during impact.</td>
</tr>
<tr>
<td>3. Place Accelerometers</td>
<td>Guides user through the process of placing accelerometers at predefined areas of a model to conform to testing standards outlined by National Highway Traffic Safety Administration (NHTSA).</td>
</tr>
<tr>
<td>4. Create Energy Groups</td>
<td>Guides users through the process of creating energy groups (a collection of affiliated parts) to measure the balance of energy in a model before and after an impact condition.</td>
</tr>
<tr>
<td>5. Nodeout Request</td>
<td>Guides user through the process of selecting certain nodes at which measures of displacement will be taken after impact.</td>
</tr>
<tr>
<td>6. Position Barrier</td>
<td>Guides user through the process of selecting appropriate barrier for test and positioning the barrier using NHTSA coordinates and user generated calculations heights of loaded vehicle.</td>
</tr>
<tr>
<td>7. Define Initial Conditions</td>
<td>Guides user through the process of defining the initial or boundary conditions of the model such as the angle of approach, friction coefficient at barrier face, velocity of impact, etc.</td>
</tr>
<tr>
<td>8. Create Contacts</td>
<td>Guides user through the process of selecting the appropriate means of creating contact algorithms between the vehicle and the barrier.</td>
</tr>
<tr>
<td>9. Position Dummy</td>
<td>Guides user through the process of selecting appropriate dummy for loadcase analysis and positioning dummy in vehicle in compliance with NHTSA testing requirements.</td>
</tr>
<tr>
<td>10. Route Seatbelts</td>
<td>Guides user through the process of routing seatbelts around dummy in compliance with NHTSA testing requirements.</td>
</tr>
<tr>
<td>11. Write Input Deck</td>
<td>Converts the graphical representation of the model into a text-based input file that can be read by the solver.</td>
</tr>
<tr>
<td>12. Generate Report</td>
<td>Automatically generates report of the simulation in an HTML format containing data of interest in tabular and chart format. Algorithms are used to sample and filter data per best practice guidelines.</td>
</tr>
</tbody>
</table>

Strut Groups used at what times. These data were arrayed into a spreadsheet that charted, by week, which engineers in each group were using which of the technology’s features. Analysis of the tracking logs indicated that engineers in both the Piston and Strut groups used a wide array of CrashLab’s features soon after the technology was implemented. But after the thirteenth week of use, engineers in the Piston group greatly reduced their use of the technology, while engineers in the Strut group converged on the use of six of its features. Consequently, I divided the data into three phases. Phase 1 comprised data collected before CrashLab was implemented. Phase 2 comprised data collected from the first through the 13th week of CrashLab use. Phase 3 comprised data collected from the 14th through the 50th week of CrashLab use.

In the second stage of analysis, observational data were used to explain why the patterns identified in the analysis of the tracking logs existed and what consequences arose from these usage patterns. To explore why the patterns existed, I coded in six steps. In the first step, I flagged all of the observation records for all instances in which informants first heard about CrashLab. Codes were then applied to each of these instances to explain (a) what the informant learned and (b) what their initial impression was. In Step 2, I flagged each instance in which informants first began to use CrashLab. I applied codes to these instances that explained (a) what they did when they first used it and (b) reasons they gave for why they used those features. In Step 3, I used the process of axial coding (Strauss and Corbin 1998) to create new codes that linked together codes from Step 1 about what informants initially heard about CrashLab to codes from Step 2 about how they used it for the first time. Step 4 was used to code each instance of CrashLab use or nonuse for each informant after his or her first use. For each of these instances, I applied codes that explained (a) whether the informant decided to use CrashLab or not and (b) reasons he or she made this decision. If the informant did use CrashLab, codes were also applied to excerpts explaining (c) how he or she used CrashLab and (d) what he or she did with the results of the simulation model produced from its use. In Step 5, I compared the coded seg-
ments from Step 4 (b–d) across the observations of each informant to identify what patterns were common within each of the groups and which patterns were different. In Step 6, I followed Glaser’s (1978, pp. 120-126) strategy for theoretical sorting to mark the activities and events in the data that led to similarities or differences in patterns of CrashLab use and nonuse across the Piston and Strut groups.

Together, the results of these coding steps of the observational data uncovered (1) what engineers believed CrashLab was supposed to do for them, (2) reasons why engineers in the Piston and Strut groups initially made similar appropriations of CrashLab’s features, (3) why, over time, the two groups veered from one another in their appropriations, (4) why and how the Strut group enacted a shared affordance and the Piston group did not, and (5) what consequences this shared affordance had for advice seeking.

The third stage of analysis was used to verify and expand the consequences of the enactment of shared affordances through the use of social network analysis. Using the data collected from each wave of the sociometric survey, I constructed adjacency matrices for the entire Piston Group and the entire Strut Group, where rows in the matrices indexed engineers who sought consultations from their colleagues, and columns indexed engineers whose consultation was sought. Each time an engineer was consulted by another colleague (in-ties), I recorded a 1 in the corresponding column. After the completion of this task, 18 separate matrices were produced for the Piston group and 18 more for the Strut group (three matrices for the model set-up and three for the model analysis consultations at each of three points in time: before implementation of CrashLab, during Period 1, and during Period 2 of CrashLab use).

All engineers in either the Piston or the Strut group worked and saw each other, face-to-face, on a regular basis; consequently, there was frequent interaction among them. However, as Barley (1990) has suggested, evidence of a strong collegial bond between workers exists when they discuss not just one, but a variety of work-related matters. To determine which ties among engineers were strong and which were weak I combined all three of the matrices for model set-up and analysis advice consultations in a given period into one valued adjacency matrix, representing all of the consultations made about model set-up or analysis activities in that period. These matrices were dichotomized (converted to binary relations) by setting each cell equal to 1 if its value was greater than or equal to the average number of consultations sought from any engineer in the group during that time period \(a_{ij} \geq \bar{r}\), otherwise the cell was set to 0. This transformation produced one directional dichotomized matrix for model set-up consultations and one such matrix for model analysis relations for each of the three time periods for each of the Piston and the Strut groups.

Because changes in the structuring of advice networks are evinced by alterations in what people talk about and to whom they talk, two complementary methods were used to analyze these relational data. The first method was to measure the actual change in density of each group’s consultations networks over time, which provided evidence of whether the topics engineers discussed (e.g., model setup or model analysis) changed. The density of a directed graph is equal to the proportion of arcs present in the graph. Because the matrix for directed data is asymmetrical (a directed line from person A to person B will not necessarily involve a reciprocated line from person B to person A) the maximum number of lines that could be present in a digraph is equal to the total number of pairs that it contains. This is calculated as the number of arcs divided by the possible number of arcs. Since an arc is an ordered pair of nodes, there are \(n(n-1)\) possible arcs. The density of a digraph is a fraction that goes from a minimum of 0 if no arcs are present to a maximum of 1 if all arcs are present. If the density is equal to 1, then all dyads are mutual (Wasserman and Faust 1994, p. 129). To execute the density analysis, a paired-sample bootstrap technique (Snijders and Borgatti 1999) was employed. The bootstrap technique computes estimated sampling variance of the means of the paired networks by drawing 5,000 random subsamples from each and constructing a sampling distribution of density measures from them. The differences in the densities of these samples are recorded after each iteration. The standard error of those differences is then used to calculate a bootstrap t-statistic, indicating whether or not the difference in the densities of the observed networks is significant. This density analysis allowed for a determination of whether changes occurred in the density (number of consultations) of the Piston and Strut group networks around model setup and model analysis talk.

The second analysis method used to assess network changes was the quadratic assignment procedure (QAP; Hubert and Schultz 1976). QAP measures change in the structure of relationships among actors. Because network data are relational, the values in each cell of an adjacency matrix are not indepen-

---

6 Generally, the bootstrap method is preferred to a classical formula for the standard error of a mean (\(s/\sqrt{n}\)) because, by treating the two networks as if they existed independently, the classical formula underestimates the true sampling variability and gives a result that is too optimistic in rejecting the null hypothesis that the two densities are the same. The procedures used to calculate the bootstrap t-statistic can be found in Version 6 of UCINET (Borgatti et al. 2002).
dent of one another, so estimation procedures designed for independent observations will calculate incorrect standard errors. QAP is a resampling-based method, similar to the bootstrap, for calculating the correct standard errors. QAP calculates Pearson’s correlation coefficient between corresponding cells of two data matrices. The procedure repeats these calculations multiple times by holding the structure of one matrix constant and randomly permuting the rows and columns of the other, thus testing if the association between the two networks is statistically significant. The QAP analyses were used to test whether the structure of those networks (from whom people sought advice, as opposed to what they sought advice about, which the density analysis reveals) also changed. Specifically, the resulting correlations should demonstrate a statistically significant correlation between the structures of the networks over time if they did not change and a nonsignificant correlation if their structures did change.

Findings

Implementing CrashLab: Exposure to Differential Frames

Three weeks before it was implemented and available for use on engineers’ workstations, classes were held in Autoworks’ corporate software training center to teach people how to use CrashLab. All engineers from the Piston and the Strut groups were required by their managers to attend. In total, six training sessions were held and engineers signed up for them based on their personal schedules. During these classes, trainers introduced engineers to the material features (which were essentially individual modules fronted by a uniform interface) offered by the technology. During the classes, engineers used generic simulation models to learn how to employ each of the 12 features. While the content covered in each of the classes was similar, the messages disseminated in them about CrashLab’s usefulness were not. The instructor who taught members of the Piston Group informed them that CrashLab was going to be useful because it would speed up the way they worked. This message linking CrashLab to faster working practices was reinforced by the Piston Group’s manager, who had attended the same meetings that the trainer did where the technology’s developers emphasized to them how it would speed up the work of engineers. The instructor who taught members of the Strut Group, by contrast, gave no specific insight into the utility of CrashLab. Neither she nor the Strut Group’s manager were briefed by the developers about what the technology was supposed to do; thus, neither passed any information on to the engineers in that group. Because the training sessions occurred a full three weeks before CrashLab was implemented, engineers in both groups had time to reflect on their training and discuss how the technology was supposed to change their work before they ever had a chance to test it on routine tasks. Additionally, engineers in both the Piston and the Strut groups attended staff meetings where their respective managers told them about CrashLab—what it was for and how it worked. Members of both groups also attended several other events at Autoworks, including computer-aided engineering (CAE) conferences and several quality assurance meetings. In each of these meetings, they were told what CrashLab was supposed to be used for and how it worked.

An examination of the talk about CrashLab that engineers heard at each of these events revealed two broad frames that portrayed the nature of the new technology in relation to engineers’ work. Those trainers, managers, and developers who had been involved in some way with CrashLab’s development created what I call an efficiency frame, to explain to engineers how CrashLab would help them to work faster and more consistently than they had before. Those trainers, managers, and developers who were not involved in the development of CrashLab created an inevitability frame to let engineers know, in very general terms, that the new technology would inevitably lead to changes in the way they worked, but refrained from giving them any specific guidance as to what those changes would be. Table 3 summarizes these two frames by describing the various messages that trainers, managers, and developers used to create them and examples of these messages they relayed to engineers during CrashLab’s implementation.

Table 4, Panel A provides an overview of the 24 events that engineers attended during which they were exposed to frames about CrashLab. The data show that engineers from the Piston Group were exposed to more messages that constituted an efficiency frame than were engineers from the Strut Group, while more PEs from the Strut Group were exposed to messages that constituted an inevitability frame than were engineers from the Piston Group. To discern whether this trend was prevalent at the individual level, I calculated the average number of messages of each type to which a PE in either the Car Group or the Truck Group was exposed. The results, presented in Table 4, Panel B, indicate that the average engineer in the Piston Group was exposed to nearly four times the number of efficiency messages than inevitability messages while the average engineer in the Strut Group was

---

7This determination was made by multiplying the number of engineers in his/her group who were present at a given event by the number of frames used in that event summed across all 24 events, and then divided by the total number of engineers in his/her particular group who were present in any of the events.
Table 3. Examples of Trainers and Managers Framing CrashLab Implementation

<table>
<thead>
<tr>
<th>Frames</th>
<th>Work Will be More Efficient</th>
<th>Work Changes are Inevitable</th>
</tr>
</thead>
</table>
| 1: Time is Lost in Tedium | *(During Corporate Training Session)*  
**Trainer:** Good morning sirs and madams. We'll go ahead and get started now. Today we're going to be talking about a new tool for crash analysis called CrashLab that was initially developed here at Autoworks. Basically, I think this is some software you're going to like because it will automatically do a lot of the parts of the job which you don't like and that are boring. This is going to give you more time since you won't be spending so much time doing the tedious activities.  
*(During Corporate Training Session)*  
**Trainer:** … It will automatically find a node. I think it’s at the center of your instrument panel on the top.  
**Engineer 1:** Are these procedures required? That means you cannot skip any steps of this?  
**Trainer:** No you can't.  
**Engineer 2:** That seems sort of annoying. That means we have to change the order of operations.  
**Trainer:** I think to use CrashLab you'll have to do things a bit differently; there's really no way around it. |
| 2: Automation Increases Speed of Work | *(During Piston Group Staff Meeting)*  
**Manager:** Apparently CrashLab is supposed to help make sure we stay ahead of the development curve. It sounds like the barrier positioning and accelerometer placement functions work really good and will save you lots of time, so make sure to make use of them.  
**Engineer:** Do you know how the algorithm works behind that, I'm just curious?  
**Manager:** Behind what?  
**Engineer:** For the automation.  
**Manager:** I don’t know, but do you remember Brett Pascal who used to work in [another vehicle program group]? I think he was involved in it some how so you could ask him. Anyway, you should be able to figure it out with the training and then we should see you work like lightening.  
*(During Piston Group Staff Meeting)*  
**Manager:** So why do we need to change what's going on now if we're getting good results?  
**Engineer 1:** Are you not happy with how we're doing things?  
**Manager:** Whoa, hold on a minute. I don’t want to change your work. I really don’t care what you do as long as you generate the results you need to evaluate your loadcases. As far as I’m concerned you can use whatever software you want. But as far as the changes go, those are driven by CrashLab, just improvements over older functions so I’m sure it won’t be a big deal, but it will require some adjusting. |
| 3: Faster Product Development Gives Competitive Advantage | *(During Strut Group Staff Meeting)*  
**Manager:** If you haven’t had a chance to go to CrashLab training yet you should because CrashLab will help us with closing this gap that Ernesto’s been discussing because it will automate a lot of your work that we are doing to increase the speed. So if we do this we can use this tool to help close this gap more.  
**Director of Safety:** Yeah, I've seen CrashLab and it looks pretty good. This is definitely the kind of technology that will help to give us a competitive advantage so we can reduce our engineering time. So definitely check it out.  
*(During Strut Group Staff Meeting)*  
**Manager:** So I know what you're thinking about using CrashLab. Why should you do it when other things work fine and it's only going to slow you down? Yeah, CrashLab might be a bear in the short run but a short productivity loss is nothing compared to better benefits in the future.  
**Engineer 1:** That's pretty dramatic there!  
**Manager:** You’re right, but I’m just saying  
**Engineer 2:** So we really need to do use this?  
**Manager:** Just scale up to it. Figure out how to use it and migrate your work over. Then it won’t be such an impediment on it. |
Table 4. Engineers’ Exposure to Discursive Frames

A. Engineers and Frames at Various Implementation Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Number of Piston Group Engineers in Attendance</th>
<th>Number of Strut Group Engineers in Attendance</th>
<th>Number of “Efficiency” Messages</th>
<th>Number of “Inevitability” Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Only to Piston Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Piston Group Staff Meeting</td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>2. Piston Group Staff Meeting</td>
<td>14</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>3. Piston Group Staff Meeting</td>
<td>9</td>
<td>0</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>4. Piston Group Staff Meeting</td>
<td>14</td>
<td>0</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Open Only to Strut Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Strut Group Staff Meeting</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>6. Strut Group Staff Meeting</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>7. Strut Group Staff Meeting</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>8. Strut Group Staff Meeting</td>
<td>0</td>
<td>14</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>9. Strut Group Staff Meeting</td>
<td>0</td>
<td>14</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Open to Both Piston and Strut Group</td>
<td>10. Training Session #1</td>
<td>11</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>11. Training Session #2</td>
<td>8</td>
<td>0</td>
<td>17</td>
<td>3</td>
</tr>
<tr>
<td>12. Training Session #3</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>13. Training Session #4</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>14. Training Session #5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>15. Training Session #6</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>16. Managerial Meeting</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>17. Managerial Meeting</td>
<td>2</td>
<td>0</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>18. CAE Conference</td>
<td>11</td>
<td>2</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>19. CAE Conference</td>
<td>2</td>
<td>1</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>20. Informal Conference</td>
<td>9</td>
<td>1</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>21. Informal Conference</td>
<td>6</td>
<td>2</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>22. Quality Assurance Meeting</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>23. Quality Assurance Meeting</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>24. Quality Assurance Meeting</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

B. Difference in Exposure to Frames by Group

<table>
<thead>
<tr>
<th>Frames</th>
<th>Number of Messages to which Average Engineer in Piston Group was Exposed</th>
<th>Number of Messages to which Average Engineer in Strut Group was Exposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Efficiency” Frames</td>
<td>13.01</td>
<td>2.29</td>
</tr>
<tr>
<td>“Inevitability” Frames</td>
<td>2.72</td>
<td>11.34</td>
</tr>
</tbody>
</table>

exposed to more inevitability than efficiency messages at roughly the same ratio. Put more simply, 75 percent of all the messages to which engineers in the Piston Group were exposed constituted an efficiency frame and 72 percent of those messages to which engineers in the Strut Group were exposed constituted an inevitability frame. These data suggest that from the earliest moments of learning about CrashLab, engineers in the Piston and Strut Groups were exposed to different information about the way that the new technology would affect their work. Understandably, engineers in the Piston Group talked amongst themselves about how CrashLab would speed up their work while engineers in the Strut Group spent their time speculating about what it would be useful for. Engineers learned about the features of CrashLab and how to operate them equally irrespective of the training session they attended.
CrashLab was implemented without incident in early September 2005, and by the beginning of October the majority of engineers in the Piston and Strut groups had experimented with its features. Members from both groups were optimistic about CrashLab’s possibilities. For the first 13 weeks after it was implemented, engineers in both groups used CrashLab regularly. Below, I discuss how engineers used the technology’s features in divergent ways that allowed them to meet the contextual demands imposed on them by their work environments. By using different combinations of CrashLab’s features, engineers created a number of individualized affordances, but no shared affordances emerged in either group during this first phase of use.

Divergent Patterns of Feature Use in Both Piston and Strut Groups During Weeks 1–13

Piston Group

Members of the Piston Group had begun to form interpretations about CrashLab as a technology that should speed up their work. Therefore, when they made decisions about whether or not to use CrashLab, they were making decisions on variables that were important to model building tasks, because such tasks were the only tasks that engineers would purposefully want to speed up. As one engineer commented, “you definitely want to make building models faster ‘cause that’s the unskilled part of the work. You want to take your time with analysis. That’s the real engineering work.” Because engineers in the Piston Group viewed CrashLab as a model technology that was supposed to enhance their speed, they often compared it to other technologies that were already well established at speeding up work. The following interaction between two engineers (E) in the Piston Group demonstrates such comparison making by judging the functionality of CrashLab against an existing technology called HyperMesh:

E1: Do you have time to run a LINCAP test for iteration 5?
E2: Are the changes to the cross car beam done?
E1: No, you have to make them.
E2: When do we need to submit?
E1: Just to have the results by Thursday.
E2: Is it set-up in HyperMesh or CrashLab?
E1: HyperMesh. CrashLab doesn’t seem to do as much for preprocessing.
E2: Yeah, that’s why I asked.

Engineers in the Piston Group were making decisions about whether to do model set-up and analysis activities in CrashLab by determining if the new technology could help them complete such tasks faster than they could do with a technology already used within the group (HyperMesh). Because engineers were exposed to information telling them that CrashLab would speed up their work, they made most of their decisions about whether and how to use CrashLab based on a criterion of speed.

Engineers often felt that using a smaller number of CrashLab’s features (as opposed to the entire package) could actually make their work faster. On one such occasion, an engineer (E) was reviewing some of the results of simulations on the configuration of a bracket with his manager (M), when the manager unexpectedly requested more information for a meeting he was to attend the next day.

M: Did you run tests putting it [the bracket] at the other angles?
E: No, but you can see how the trend is looking.
M: Is it consistent?
E: Yeah, it follows the trend so it will just show that the offset gives you a worse pulse.
M: Can you give me just some numbers on that or some charts to show when I meet with [the design team] tomorrow, ‘cause you know they’ll ask?
E: You just want what the numbers are?
M: Yeah, just to show them.

When the engineer returned to his desk he commented that if he ran the simulation over night for the extra iterations that his manager requested, he would not have enough time to assemble all of the data into tables and charts for his manager’s meeting. The engineer then thought of a quick solution.

Well, tomorrow I can just open it up in CrashLab and it will give, I think, most of the numbers [the manager] wants in that report. I don’t think it’s enough to be conclusive but it should be enough to show the trend and it will be fast to do that. That will be good to see if it works for that or not.

Following this logic, the engineer opened a text-editor file and made the changes to the angle of the bracket there, as opposed to using CrashLab’s features to make this change, and then used a script borrowed from one of his colleagues to submit the job to the solver, again instead of using CrashLab’s features to perform this activity. The next day when the solved model was returned to his workstation he used CrashLab to generate a report based on the data that he submitted to his manager. Thus, the immediate demands of the situation,
coupled with his understanding of CrashLab as a tool to speed up work prompted the engineer to use only one feature of CrashLab to make the desired changes rather than running the model through the program from start to finish. In such an instance, use of CrashLab’s features allowed the engineer to enact an individualized affordance that enabled him to create a brief report.

**Strut Group**

Engineers in the Strut Group were also motivated to use CrashLab to meet immediate demands placed on their work. However, they had much less specific interpretations of its utility than their counterparts in the Piston Group. When making determinations about how to use CrashLab, engineers were not looking for specific features that would allow them to speed up their work. In other words, when an engineer was faced with a contextual demand, she did not have to first determine if using CrashLab would be faster than using another application; instead, she just had to decide whether CrashLab provided sufficient functionality to accomplish the task at hand. Often, multiple engineers made distinct assessments about what demands CrashLab could successfully meet, as this interaction between two engineers indicates:

**E1:** Did you figure out if you get a better pulse if you up-gauge the rails?
**E2:** Yeah, I ran a frontal impact test and it was better at 3 mils.
**E1:** Oh.
**E2:** I up-gauged it in CrashLab.
**E1:** Huh. I’ve never used it [CrashLab] to do that. I just tried it to route the belt for the dummy.
**E2:** I never used it for that [seat belt routing]. I just go into the text editor to do it.
**E1:** Oh, that might be slower but it might be more accurate, I guess. I never thought about using it for that.

In many cases, engineers in the Strut Group decided to use one of CrashLab’s features knowing full well that it would be faster to use another technology to perform the same task. In these situations, the demands of their immediate context helped them to view CrashLab as the best choice for the job and differential demands meant that engineers often used different material features, thus enacting individualized affordances.

The following exchange between an engineer (E) and a designer (D) who were discussing the placement of a battery tray in the engine compartment further illustrates the types of situations that led engineers in the Strut Group to choose to use CrashLab:

**E:** If [the battery tray] could just be like 4 centimeters higher I think that would take care of it.
**D:** I don’t think we can move it higher without interfering with the reservoir. Could we move it back a few inches?
**E:** Maybe, I’m not sure. I’ll do some tests on it to see. How much room is there?
**D:** To move it back?
**E:** Yeah.
**D:** We could probably go like three inches or so

When the engineer returned to her desk she pivoted toward me and said,

I think maybe what I’ll do is create another energy group to see if we get a better pulse if we move it back like he said. I’m not sure exactly what group, but [pauses] I think you can do that in CrashLab. I mean it will select if for you. Maybe I’ll explore that.

This particular engineer made the decision to use one specific feature of CrashLab in her work because its automated routines would perform a task that she did not feel she had the knowledge to accomplish correctly on her own. In this instance, she enacted an individualized affordance enabling her to take section forces from the model. The calculus of decision making for this particular engineer and for most engineers in the Strut Group was divorced from any preconceived notions about how CrashLab should be used. Alternatively, whenever CrashLab’s features were seen to confer some particular advantage for the user, those and only those features were used.

In summary, immediately after CrashLab was implemented, engineers in both groups began to explore various ways of using it by combining its features in unique ways. Divergent patterns of feature use within each group indicated that engineers were enacting a number of individualized affordances.

**Convergent Patterns of Feature Use in Strut Group Only During Weeks 14–50**

At the end of the 13th week after CrashLab was implemented, both the Piston and the Strut groups reached a key stage-gate in Autoworks’ rigid vehicle development process: the virtual mule vehicle assessment (VMVA). At this critical juncture, Autoworks’ senior management made a complete assessment
of the viability of the vehicle program by evaluating the results of performance tests on a virtual mule vehicle. As engineers commented, work after VMVA became more serious and deliverables to management began to occur with increasing frequency. In fact, the number of iterations of a particular design that engineers had to build and analyze increased after VMVA and continued at a high level until the next stage-gate. Thus, after this point one might expect that engineers would have less incentive than before to explore CrashLab’s features. In the course of observations, I noticed a decisive change in the way CrashLab was used within the two groups after the VMVA stage-gate. Yet, the nature of that change differed between them.

**Piston Group**

Engineers in the Piston Group sensed the urgency of more frequent deadlines and grew concerned that if they continued to use CrashLab, which they had determined during the previous three months was slower than other technologies they could use to set up models, they might fall behind schedule. The following exchange between the Piston Group manager and one of his engineers demonstrates this perceived concern:

**M:** [Looking at test results on a printed sheet of paper] It looks fine but we’ll need to run a few more tests at different angles now that we’ve had to take so many KGs off.

**E:** Ok, I’ll submit them tonight and they should be ready by Thursday.

**M:** We’ve gotta get them quicker because I have the team meeting on Wednesday afternoon.

**E:** Ok, I’ll see if Rachel [a fellow Piston Group member] can help.

**M:** Fine. And are you configuring the iterations in CrashLab?

**E:** No, I can’t if you want them that fast.

**M:** Ok, too bad. But whatever, just get ’em in.

While building and analyzing their models, engineers no longer experimented with CrashLab. In fact, they did not appear to use it nearly as frequently as they had before the VMVA stage-gate. Because they determined that (1) CrashLab was supposed to speed up their work, (2) it was not good at speeding up their work, and (3) they had to get work done faster after VMVA than before, engineers in the Piston Group dramatically reduced their use of the technology. As one engineer commented, “Once we made it through VMVA, my deadlines started coming more frequently so I just needed to get things done quicker so I went back to using HyperMesh more.” However, the engineers in the Piston Group did not entirely abandon CrashLab. Those who continued to use it after VMVA did so in much the same way that they did before the stage-gate. There were still various occasions in which they found themselves facing contextual demands that encouraged them to enact individualized affordances. Their use of CrashLab’s features remained idiosyncratic such that different engineers continued to use its in divergent ways. One perceptive engineer commented that CrashLab’s fate in the Piston Group was not out of the ordinary:

CrashLab…was ok, but not that great of an improvement. Some people still use it; but they kind of use it haphazardly, like for this or that, and that includes me. So no one is using it consistently even for themselves and not that many people are using it all together. Like, for example, I just use it from time to time to set up barriers. But some other people were using it to define contact and some other people even were using it to do nodeouts. This happens to a lot of technologies we get. They just don’t get used by enough of us consistently so they just sort of fall by the wayside.

**Strut Group**

Over the 30 weeks during which Strut Group engineers were observed using CrashLab after VMVA, it became clear that they did not make any strategic decision that the technology was good for some things and not for others. Instead, they responded to the urgency imposed by the latest stage-gate by simply ceasing to experiment with CrashLab’s features. As one engineer commented,

I didn’t really experiment with CrashLab so much after we started to get deeper in the program since there wasn’t so much time. You just do what works, you know, for those things that I wasn’t too comfortable to do anyway. You just use it in the way that everyone else is to get your job done.

In other words, engineers dispensed with the idea that they would try CrashLab out on several different aspects of building and analyzing a model just to see how it worked. They traded this exploration for calculated usage of the technology for tasks that they knew (from their experimentation before VMVA) it was capable of doing well. Without a rubric against which to measure CrashLab’s utility (e.g., the Piston Group’s belief that it should be “faster” than other technologies), engineers in the Strut Group were left free to determine that if CrashLab did something better than another technology, they would use it. Rather than interpret “better”...
narrowly as “faster,” engineers in the Strut Group normally interpreted it to mean “more accurate” or “easier.” Consider the following responses given to the question, “Why are you using CrashLab to do this task?”

E1: I sort of tested it for model checking and it caught more penetrations than just doing a 0-millisecond run. So I think it’s maybe more accurate.

E2: It’s just easier to set up the barrier with it because you don’t need to look at any standard work guidelines.

E3: I like it to set up models because it just seems to do it easily and then you don’t have to worry if you forgot this or that because all those routines are counted for in the software.

These responses indicate that engineers considered CrashLab to be useful regardless of whether or not it was faster than other tools.

By reducing their experimentation with CrashLab, engineers in the Strut Group also appeared to reduce the number of features they used. As I moved from observations with one engineer to observations with others, it became apparent that all of the engineers in the Strut Group were converging upon a common pattern of feature use. That is, each engineer no longer enacted different individualized affordances; instead, engineers across the group began to enact one shared affordance through a convergent pattern of feature use that gave them the ability to compare their outputs with each other. This convergence on a certain set of features was made possible, initially, by the Strut Group’s continued use of CrashLab after VMVA. Although continued use was necessary for convergence to occur, it was not sufficient. As engineers were using CrashLab to set up their models and shared work back and forth, they found it easier to transfer tasks and interpret the outputs of their reports if everyone was setting up the models in uniform ways. CrashLab, although not the fastest way to set up a model, did at least assure that a model was set up uniformly even if it was set up by different users. This uniformity afforded any engineer in the group the ability to post-process a model set up by any other engineer because he knew exactly the steps that were taken. I routinely observed engineers commend each other for using CrashLab to set up models. For example,

E1: I ran some of the pulses for the [vehicle model] with the revised cross car beam and they looked good.

E2: Good even with the initiator in the rail?

E1: Yeah it was fine. We’re in the range.

E2: Ok.

E1: Thanks for doing it in CrashLab; it was easier to pull the nodes.

E2: Yeah? Good.

These complementary interactions reinforced for engineers the importance of using CrashLab to set up their models so that others could easily work with them later.

There were also interactions in which engineers did not use CrashLab; the tone of these interactions was much different. Consider, for example, the following interaction between Engineer 1, who came to Engineer 2’s desk to ask a question about a model:

E1: I tried to do iterations 45–52 like you wanted but I’m getting weird results.

E2: Ok, show me. [They walk to E2’s desk. E2 sits and opens the energy curves he was looking at before he went to E1’s desk.]

E1: See? [E1 points to the screen at the inflection point on the curve.]

E2: That looks weird. Hmm.

E1: Where did you take the J and K heights?

E2: Can you pull the model up? [E1 opens the model and rotates it to an isometric view. He rotates the model again to see the under-carriage. E2 points to a location on the left rail just past the kick down.] Here. I took it here.

E1: That’s the problem, I think. That’s not the right spot. Why did you take it there? Did you use CrashLab?

E2: No, I did it in [names another system].

E1: You should really use CrashLab because it will put it in the right spot for you. You don’t even have to select if yourself—that way you won’t make that mistake. It’ll make it easier if you want someone else like me to do this for you. I use the model check all the way through the barrier position features each time. You need to be doing that too, probably.

In interactions such as this, engineers reprimanded one another if they were not using CrashLab to set up models. More importantly, as the last line of the excerpt indicates, E1 tells E2 not only to use CrashLab; he tells him how to use it. Engineers frequently made announcements about which of
CrashLab’s features they were using and encouraged others to use the same ones.

In summary, engineers in the Strut Group also felt the pressures of more frequent and difficult deadlines after VMVA. Unlike their counterparts in the Piston Group, however, Strut Group engineers did not perceive CrashLab as an impediment to meeting those deadlines. Because they never formed the impression that CrashLab was supposed to speed up their work, they did not anticipate that it would, and, alternatively, they were not disappointed when it did not. Instead, they were left to their own devices to determine what kind of technology CrashLab should be.

Comparison of Crashlab Feature Use Across Time Periods

The observational data presented above suggest that before the VMVA stage-gate, engineers in both the Piston and Strut groups were using CrashLab’s features in divergent ways. The data also suggest that after VMVA, engineers in the Strut Group only (not the Piston Group) began to converge on a common set of features that would collectively afford them the ability to compare the results of their work. Drawing on the data obtained from informants’ tracking logs, I examined the number of CrashLab’s features (0–12) that engineers used after implementation along with the number of engineers in the Piston Group (19 members) and the Strut Group (17 members) who could have potentially used them. The graphs in Panels A and B in Figure 1 indicate that in both groups the number of features that engineers used began to drop off around the 13th week after CrashLab was implemented—just at the time of the VMVA stage-gate. As engineers began to move into a stage on the vehicle program in which deadlines occurred more frequently and they were expected to provide results in a timelier manner, they began to experiment less with CrashLab’s features. Although the VMVA stage-gate appears to have been an important turning point for both the Piston and the Strut groups, the ways in which engineers treated CrashLab’s features thereafter was markedly different.

By comparing the graphs in Figure 1 we see that after week 13 engineers in the Piston Group began to use a much smaller number of CrashLab’s features. In fact, by week 33, engineers in the Piston Group were either using a very small number of CrashLab’s features in their work or no features at all. By contrast, engineers in the Strut Group did not significantly reduce their use of CrashLab after week 13. Although the number of features that they used in their work began to decline after VMVA, by week 20 the Strut Group was consistently using about five features on a regular basis. This trend continued until I stopped collecting logs at week 50, at which time the average number of features used by the engineers in the Strut Group had dropped only slightly, to four.

Because the VMVA stage-gate marked an important juncture in crashworthiness engineering work, I used this exogenous force as a breakpoint to demarcate two phases of CrashLab use after implementation. Phase 1 began with the implementation of CrashLab in both the Piston and the Strut groups and continued through week 13. Phase 2 began at week 14 and extended through week 50. By dividing use of CrashLab into these two periods, I was able to construct Table 5, which helps to explain the contours of the changes in CrashLab use captured in Figure 1. The data indicate that in Phase 1, engineers in the Piston Group and the Strut Group were using nearly the same number of CrashLab’s features; engineers in both groups had each experimented with 11 of CrashLab’s 12 features. Although the data indicate that engineers used a breadth of features in Phase 1, they show little depth of use. Less than half of the engineers in either the Piston or the Strut group had experimented with more than five of CrashLab’s features. In other words, Phase 1 is marked by a widely divergent pattern of CrashLab use in which engineers were using CrashLab’s features selectively, not collectively as a bundled set.

During Phase 2, engineers in the Piston Group only used 5 of CrashLab’s features as compared to the 11 features they used in Phase 1. Although they reduced the number of features they used, it did not appear that those engineers who were still using these five features were using them in any consistent way. Instead, engineers in the Piston Group were still using CrashLab features in divergent ways. Some were enacting individualized affordances for model checking while others were enacting individualized affordances for seat-belt routing. For engineers in the Strut Group, by contrast, feature use in Phase 2 looked different than Phase 1. In Phase 2, Strut Group engineers consolidated the number of features they used in the second period to six main ones. In fact, when examining the number of people who were using these features, it appears that engineers in the Strut Group did not maintain a divergent pattern of use; instead, engineers collectively converged on the use of a particular set of CrashLab’s features.

Shared Affordances and Advice Network Change

If CrashLab did induce change in the patterns of advice seeking among engineers only after they converged on the use of a common set of features that allowed them to enact a shared affordance to compare their work with each other, the
densities of the networks before its implementation and in Phase 1 should be nearly identical for model set up and analysis activities for both the Piston and the Strut groups. However, between Phase 1 and Phase 2, network densities should show changes for only the Strut Group because it is at this time that they began to converge on common appropriation of CrashLab’s features to enact a shared affordance for comparison, allowing them to increase the frequency with which they consulted with one another about model analysis activities and decrease the frequency with which they discussed model set up activities. Additionally, there should be no change in the Piston Group networks between Phase 1 and Phase 2 because they never converged on a common set of features and never enacted a shared affordance that would enable such network change.

If the densities of the networks did change across phases, one would also suspect that the structure of communication among ...
Table 5. Average Number of Engineers Using Specific CrashLab Features in Each Phase

<table>
<thead>
<tr>
<th>Feature Used</th>
<th>Phase 1 (Weeks 1-13)</th>
<th>Phase 2 (Weeks 14-50)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Piston Group</td>
<td>Strut Group</td>
</tr>
<tr>
<td>Check Model</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Create Sections</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Place Accelerometers</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>Create Energy Groups</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Nodeout Request</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Position Barrier</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Define Initial Conditions</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Create Contacts</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>Position Dummy</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Route Seatbelts</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Write Input Deck</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Generate Report</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>

Engineers would as well. Because density is a measurement of how many nodes in a network are connected (how many cells in an adjacency matrix are filled) rather than how many in-ties are present for any given actor (what the value is for each cell), a significant change in network density would mean either that engineers who did not consult one another previously were now communicating (if the network density increased) or engineers who had once consulted each other now ceased to communicate (if the network density decreased). The results of the bootstrap t-tests, which measured changes in network density, and the QAP analyses, which measured changes in network structure, for the Piston Group are presented in Table 6 and the results for the Strut Group are presented in Table 7. The data show that in the Piston Group, neither the densities of the model set up nor the model analysis consultation networks changed significantly across the three time periods. Further, the results of the QAP analysis show a highly significant correlation between the structures of the model set up and analysis networks across the three time periods. These findings indicate that CrashLab did not bring about any change in the Piston Group’s advice network. One year after the new technology was implemented, the Piston Group remained an organization heavily dominated by the practice of routine model set up as opposed to detailed model analysis.

The results of the density and QAP analyses for Strut Group tell a different story. The densities of both the model set up and analysis networks showed no change in patterns of consultation from the time before CrashLab was implemented to the end of Phase 1. Similarly, the QAP analysis shows that the structures of the networks in these periods were strongly correlated with each other, further illustrating that no change had taken place. In other words, the mere implementation and diffusion of CrashLab had no independent effect on the Strut Group’s advice network. According to the theory outlined above, convergence in the use of CrashLab’s features in Phase 2 afforded engineers the ability to work in the same way as their group members and, therefore, to consult each other therefore, to consult each other about analysis activities. If this theory is supported, the data should show a significant difference in densities of the consultation networks between Phase 1 and Phase 2 and also demonstrate that the structure of the consultation networks differed from one another between these two phases. The data in Table 7 show precisely these results. The only significant change in densities in the model set up and consultation networks occurred between phases 1 and 2. Similarly, the only change in the structure of these networks also occurred during phases 1 and 2.

As one example of how converging on a shared appropriation of CrashLab’s features enacted a shared affordance of output comparison in the strut group, consider the following advice seeking consultation around changes in geometry. An engineer (E1, below) in the Strut Group was faced with a dilemma in that he could not achieve sufficient energy dissipation in the front rail of the vehicle without changing the gauge of the rail. However, the weight restrictions for the vehicle, which were set by the lead vehicle architect, prohibited him from increasing the thickness of the rail by more than 0.1 mm. The engineer, who needed an increase of at least 0.4 mm to achieve an acceptable level of performance sought out another engineer on the team (E2) who he knew had recently dealt with a similar problem for a different kind of barrier test:
### Table 6. Comparison of Densities and Structures of Piston Group Advice Networks Over Time

<table>
<thead>
<tr>
<th>A. Set Up Advice Networks</th>
<th>Before Implementation</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density Difference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.03 (.49)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>-.04 (-.61)</td>
<td>.03 (.43)</td>
<td>1</td>
</tr>
<tr>
<td>QAP Correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.45*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>.39* (.65)</td>
<td>.91*</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. Analysis Advice Networks</th>
<th>Before Implementation</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density Difference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.03 (.65)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>-.02 (-.34)</td>
<td>-.05 (-.97)</td>
<td>1</td>
</tr>
<tr>
<td>QAP Correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.67*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>.63* (.86)</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

*Numbers in parenthesis are t-tests for corresponding parameters. *p < .01

### Table 7. Comparison of Densities and Structures of Strut Group Advice Networks Over Time

<table>
<thead>
<tr>
<th>A. Set Up Advice Networks</th>
<th>Before Implementation</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density Difference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.04 (0.64)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>0.45 (7.81)*</td>
<td>.41 (6.54)*</td>
<td>1</td>
</tr>
<tr>
<td>QAP Correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.74*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>.01</td>
<td>-.01</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. Analysis Advice Networks</th>
<th>Before Implementation</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density Difference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>-.06 (-.95)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>.45 (-6.96)*</td>
<td>-.38 (-6.31)*</td>
<td>1</td>
</tr>
<tr>
<td>QAP Correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Before Implementation</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase 1 of CrashLab Use</td>
<td>.50*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Phase 2 of CrashLab Use</td>
<td>-.03</td>
<td>-.05</td>
<td>1</td>
</tr>
</tbody>
</table>

*Numbers in parenthesis are t-tests for corresponding parameters. *p < .01
E1: So how did you dissipate the energy without changing the gauge or lengthening the crush space?

E2: This will probably help you and I can make a better picture later on. It’s going to start bending here. [He points to the location of the controlled bend on the rail of the computer model on his screen (see Figure 2).] What we did was we put some initiators in there so we could get controlled bending at this point.

E1: So, where did you transfer the energy?

E2: What you want to do is take the load off of that path and transfer it somehow to the rocker. So, in the plan the rail does this [he points again to the point of controlled bending], and we have a shock tower [points] and there is a wheel there [points]. Then it hits up against the rocker. So, for here [he points to where the wheel contacts with the rocker] I’m transferring the load to the rocker. It just cannot load this. It becomes a hinge…

E1: Do you think it would work in my model?

E2: Well what boundary conditions did you use when you set it up?

E1: I just used CrashLab to do it.

E2: Ok, I did too. So you should definitely be able to use this same solution.

This ingenious solution leveraged only a slight change in the geometry of the front rails to use the natural interaction of parts in the stack-up to enhance performance. As the advice-seeking engineer commented after this meeting,

That was a pretty cool solution. I’m glad I asked. I don’t know if that will exactly work for what I’m doing, but I think I could basically use the same principle for the controlled bend to transfer the load. I’m pretty impressed. That was good thinking. I knew he used CrashLab on it, which is why I asked him. ‘Cause that means if it worked for him it would work for me.

Such advice-seeking consultations around issues of model analysis were enabled because Strut Group members enacted a shared affordance for comparison through their use of CrashLab. Engineers in the Strut Group recognized that CrashLab afforded them this capability and commented that in addition to learning important design solutions that allowed them to build better vehicles, they enjoyed talking with others about design and analysis because it forced them to think critically about their data and to transfer numbers generated in plots and graphs into three-dimensional solutions with steel and plastic.
Discussion

This paper began by laying out a tentative theory of when technology use may lead to advice network change that is meaningful for organizations. The theory suggested that network change is likely to occur only after a group of users appropriates the features of the technology in similar ways. The shared affordance is the proposed mechanism linking convergent usage patterns and advice network change. I sought empirical support for this theory through an explanatory case study of two groups of automotive engineers using the same new technology for crash test automation.

The findings showed that, initially, engineers in both groups went through a period of experimentation with the new technology. Engineers appropriated different combinations of the technology’s features and, consequently, enacted a number of different individualized affordances. Such patterns of divergence persisted in both groups for a full 13 weeks after using the new technology when both groups began a new stage in the vehicle development process and had to be more judicious in their use of the tool. In response to this stage-gate, engineers in the Piston Group began to diminish their use of the technology while engineers in the Strut Group coordinated their actions to begin using the technology in similar ways. Enacting a shared affordance for comparing simulation outputs allowed engineers in the Strut Group to talk more about analysis with each other than they had before, which led to an overall change in both the content and structure of their consultation networks. Members of the Piston Group, who never enacted a shared affordance, continued to use the technology’s features in distinct ways and never realized the type of network change that would have been needed to test this assumption, as current studies find that because technologies are comprised of many features, an understanding of feature use may help to explain particular changes to work and communication that result within organizations. Although there are many calls for the importance of studying feature use, there are few theoretical statements about how or why feature use patterns may bring about organizational effects. Markus and Silver (2008) offer one of the few such statements, arguing that using certain features may afford people particular functionalities that, in turn, allow them to change their work. The findings of this study generally support this idea, but add a few important caveats. One caveat is that affordances can be enacted differently by various users. Because there are multiple features available—each providing their own capabilities—using some of them, or creating certain combinations of those features, will provide users with distinct affordances. Not everyone in a workgroup or department may enact the same affordances and, to the extent that their work is highly interdependent, they may not be able to coordinate their outputs effectively because they are each using the technology differently. The findings of this study have shown that to link feature use to organizational changes through affordances may require a group level, as opposed to an individual level, of analysis.

In this study, both groups of engineers initially appropriated the technology’s features in idiosyncratic ways such that they enacted individualized affordances to help them complete their work, but which did not help them to interact differently with the other members of their group. During this time, the advice networks changed in neither group. It was not until members of one group converged on a common pattern of usage patterns and advice network change. I

The findings suggest that the interdependence of work among engineers in both groups made it difficult for any new technology to afford changes to the network unless most group members were afforded the same resources to use in their work. For this reason, a shared affordance, as opposed to a collective affordance, was an important antecedent to network change. Indeed, as Wilkin (2009, p. 189) observed in her multilevel study of packaged software implementation, “Often groups of users work together, interaction in their use of a system to produce outputs that have been collectively generated and upon which they are collectively reliant.” Had the group members worked on solitary tasks that needed only to be pooled upon their completion, it is possible that convergence in patterns of appropriation would not have been necessary to produce network change. Instead, the group could have adopted a configurational usage structure and enacted a collective affordance. Future empirical work is needed to test this assumption, as current studies find that shared versus configurational usage structures at the group-
level can lead to different kinds of outcomes at the organization level (e.g., Kane and Labianca 2011). Regardless, the findings do show that some group-level affordance, as opposed to simply an individualized or “functional affordance” (as Markus and Silver called it) is necessary to produce change that cascades beyond the individual or group to the organization level of analysis. Accordingly, if affordance is to be specified as a mechanism that links technology feature use to network change in organizations, some treatment of affordances as a group-level construct is needed.

Another facet this study brings into relief is that affordances are not out there waiting to be utilized, as some design science researchers claim (e.g., Norman 1990), but rather that affordances must be enacted in practice through particular patterns of feature use at the group level. That is to say, in the context of network changes in organizations, technologies can only become resources for change once group-level affordances are produced. Feldman (2004, p. 296) encapsulates this idea of resources as things that are made as opposed to given in her concept of resourcing: “resourcing is the creation in practice of assets...such that they enable actors to enact schemas. Thus, children must be turned into students to enable teachers to enact teaching schemas.” Following this logic, we might say that technologies can only be turned into resources that people can draw on to change their networks once group-level affordances are enacted. Examining network change as a consequence of how the features of a technology are used may compel researchers to rethink the role of technology as an exogenous agent of change. The data presented above showed that the implementation of the technology was, by itself, incapable of bringing about any network change. It was only after people converged on a common use of its features that the technology became a resource that could be used to change patterns of communication. Thus, while the acquisition of new resources might normally be considered catalysts for network change (Arya and Lin 2007), it is not certain that a newly implemented technology can be considered as a resource for this purpose until its use is negotiated in the shared interactions occurring amongst group members. Therefore, in the context of network theorizing, authors who take for granted that new technologies are available organizational resources for change (e.g., Eisenhardt and Martin 2000; Mata et al. 1995) may need to be more specific about whether or not the conditions have been met (shared affordances have been enacted) to turn a technology specifically into a resource for network change.

This study also responds to calls for more research into post-adoption behaviors (e.g., Jasperson et al. 2005; Lapointe and Rivard 2005). In contrast to other comparative studies of post-adoption behavior, which have shown that multiple groups began using the technology differently almost from the day it was implemented (e.g., Barley 1990; Edmondson et al. 2001; Robey and Sahay 1996), perhaps the most striking finding in this study was that both the members of the Piston Group and the Strut Group were using the newly implemented technology for nearly three months in the same fashion—individuals diverged from one another in the features they appropriated—before members of the two groups began to embark upon different patterns of use. In this case, both the initial patterns of use and the evolution of those patterns (at least in the Strut Group) were greatly influenced by the frames about the new technology that users were exposed to before implementation. In other words, pre-implementation frames can have important repercussions for post-implementation behavior. This finding highlights the importance of attending to the creation of frames about technology use (e.g., Davidson 2002, 2006). Sometimes, pre-implementation frames about a new technology are strategically crafted by powerful actors (Edmondson 2003), other times they emerge organically within groups based on norms, values, and goals (Orlikowski and Gash 1994). In either case, frames that limit or place restrictions on the kinds of task-based changes that technology users can expect (as happened in the Piston Group) may lead to delayed technology resistance and eventually stymie important network change. Frames that are sufficiently vague (as were those heard by the Strut Group) may provide users enough flexibility to search for the kinds of uses of the technology that make it useful in their work. Although these speculations, based on the data presented in this study, certainly need qualification, the evidence presented herein clearly points to the important role that frames play in post-adoption activities. Thus, frames should be carefully crafted and their uptake carefully monitored since it seems that they do shape the way people experience a technology’s features.

**Conclusion**

Explaining the link between technology implementation and network change continues to be a pressing concern to scholars and managers alike. So often, new technologies are implemented and fail to bring about the kinds of changes to work that were envisioned by their champions. To date, one explanation missing in this link has concerned when technologies are likely to bring about network change. This question of when change will occur is as important as questions about how the change process unfolds because organizational members may continue to use a technology for many years and see no change in their work and communication practices or change may begin to occur many months following imple-
mentation after those who implemented the technology suspected that it never would. The findings in this paper offer one way of understanding when technology use will bring such changes. The focus on technology feature use directs attention to what kinds of affordances for change groups might be able to realize. These concepts may be useful tools for explaining when change is likely and why it occurs sometimes but not others. Getting people to simply use a new technology is not enough to bring about network change in organizations. How they use the technology matters.

**Acknowledgments**

I thank Steve Barley, senior editor Ola Henfridsson, and members of the review team for comments that helped to improve this manuscript. Support for this research was provided by National Science Foundation Grant ITR-0427173 and a fellowship from the Alfred P. Sloan Foundation, along with generous funding from the Robert and Kaye Hiatt Fund at Northwestern University.

**References**


**About the Author**

Paul M. Leonardi is the Pentair-Nugent Associate Professor at Northwestern University. He teaches courses on the management of innovation and organizational change in the School of Communication, the McCormick School of Engineering, and the Kellogg School of Management. His research focuses on how companies can create organizational structures and employ advanced information technologies to more effectively create and share knowledge. This focus draws attention to the processes through which technological artifacts and informal communication practices coevolve.