Organization Science
Vol. 22, No. 2, March–April 2011, pp. 347–369
ISSN 1047-7039 | EISSN 1526-5455 | 11 | 2202 | 0347

DOI 10.1287/orsc.1100.0529
© 2011 INFORMS

Innovation Blindness: Culture, Frames, and Cross-Boundary Problem Construction in the Development of New Technology Concepts

Paul M. Leonardi
Department of Communication Studies, Department of Industrial Engineering and Management Sciences,
Northwestern University, Evanston, Illinois 60208, leonardi@northwestern.edu

This paper has three goals. The first is to understand why members of one organizational department are blind to the reasons why members of another department do not share their ideas for a new technology—what I call a "technology concept." The second is to understand what consequences this "innovation blindness" has for the development of technology concepts across organizational and occupational boundaries. The third is to uncover strategies organizations might use to successfully develop a new technological artifact from the technology concept even if innovators never understand the nature of their own blindness. To achieve these goals, I draw on research on organizational cultural toolkits to construct a framework suggesting that technology concepts frame cultural resources, which are then used to construct the very problems the technological artifact will be built to solve. From this perspective, culture does not directly shape technological artifacts. Rather, a technology concept activates culture as it draws frames around resources that will guide people's problem construction practices. By acting as a frame through which problems can be constructed, technology concepts play a key role in selecting the set of cultural resources that will be used to develop technological artifacts. I explore this framework through a qualitative study of computer simulation software in a major automotive engineering firm.

Key words: innovation; boundaries; technology concept; ambiguity; frames
History: Published online in *Articles in Advance* June 21, 2010.

It was six men of Indostan
To learning much inclined,
Who went to see the Elephant
(*Though all of them were blind*),
That each by observation
Might satisfy his mind.

The opening lines of John Godfrey Saxe's (1873, pp. 77–78) famous poem, drawn from an ancient Hindu fable, introduce six blind men who wish to know an elephant. As the poem goes, each man touches a different part of the elephant and each describes the animal differently: like a wall, a spear, a snake, a tree, a fan, and a rope. The catch, of course, is that because the men are blind they cannot see that the part of the elephant they have touched is different than the part touched by their friends. After hearing the others' descriptions of the elephant each man comes to believe that only he has correctly ascertained the animal's true nature—the others have not—and conflict begins.

A body of research on innovation processes implicitly suggests that new technology development efforts frequently operate in much the same way as blind men touching an elephant. Researchers have shown that individuals from different cultural backgrounds (e.g., occupations, departments, etc.) participating in the development process often have distinct ideas of what the technology's functionality—what it is capable of doing—should be (Callon 1980, Van de Ven 1986). When these groups come together, they disagree about what features the technology should have and their disagreement impedes the development process (Dougherty 1992). Although it has been shown that some amount of disagreement about a new technology's functionality is useful for producing a better innovation (Amabile 1996), the evidence suggests that insurmountable disagreements result in delays, suboptimal compromises, and, most often, failure to ever build a working technology.

But just because a person from one cultural background sees differently than someone from another, is the person necessarily blind to the views of her compatriot? Research has shown that innovators can sometimes accurately recognize the vantage points of others and that they often try to find ways to translate their cultural differences into a common vocabulary that can be used to bridge them (Bechky 2003, Carlile 2002). Contrary to such findings, the story of the blind men and the elephant depicts a circumstance where actors never have the opportunity to resolve their differences because they are blind to the reasons why others would perceive an alternate technology. This paper explores why members of one organizational department are blind to the reasons why others do not share their view of what the functionality of a new technology should be. To do so, it seeks to understand what consequences this "innovation blindness" has for the development of new technologies and to uncover strategies that organizations might use to successfully develop a new technology even if innovators who work in them never understand the nature of their own blindness.

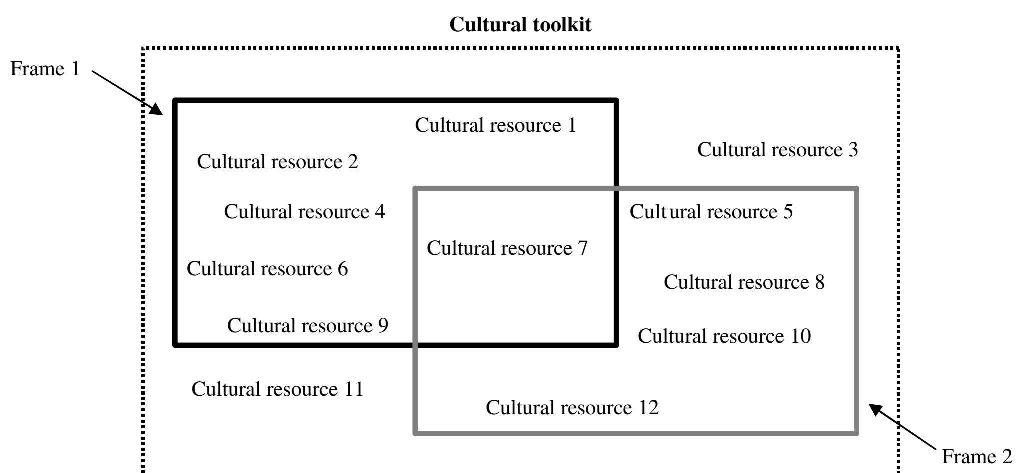
Theoretical Background

Cultural Resources and Frames

Scholars have suggested that disagreements between groups involved in technology development can be traced to cultural differences (Ettlie 2007). Although such a claim is no doubt accurate in its broad implications, researchers in communication studies argue that treating culture as a causal variable inappropriately advocates a view of it as a "thing" that exists independent of and apart from people (Alvesson 2002, Eisenberg and Riley 2001). An alternative perspective is to treat culture as a set of resources that individuals draw on to enable their everyday actions. Swidler's (1986, p. 280) widely used notion of the "cultural toolkit" exemplifies this perspective by suggesting that "culture provides the materials from which individuals and groups construct strategies of action." In such a view, culture is neither ubiquitous nor unilateral. Rather, it is made manifest in a combination of resources (e.g., skills, goals, attitudes, theories) that can be variously combined and differentially employed in unique situations. Culture is activated in action. As a collection of available resources, organizational cultures are open to reinvention and the elements that constitute them can be recombined, although some consorts of resources are mobilized more frequently than others, thereby gaining stability and longevity.

Swidler suggests that there are always more cultural resources available to any one person than they can use at a given time. Imagine multiple actors who all have access to the same store of cultural tools. A specific job requires only some of the available tools in the store, not all of them. Actors assemble specific toolkits for the job at hand from the multitude of cultural tools available in the store. Thus, an important question concerns how it is that they select the ones that will guide their action. To answer this question, many scholars have drawn on the concept of "frames." For Goffman (1974), frames demarcate a set of cultural resources that one will use to create schemata of interpretation that guide her actions (cf. DiMaggio 1997, Snow et al. 1986). The concept of frames helps to explain why groups of people who share access to the same toolkit of cultural resources might act and perceive the world in different ways. Consider, for example, the illustration presented in Figure 1. A cultural toolkit (represented by the perforated box) consists of multiple cultural resources. Frames (represented by the black and gray boxes) group certain resources together. Resources and frames are not exclusive. Some resources (such as resource 7) may be included in multiple frames, whereas others (3 and 11, for example) are not included in any. Although the same cultural resources may be available to everyone, the arrangement of those resources into particular frames may result in differential strategies for action across the organization. Such a view follows research suggesting that different groups within that broad organizational culture can share strong similarities while evincing clear differences in their strategies for action (Trice 1993). Research suggests that diverse bases of scientific knowledge, organizational structuring decisions, and skill requirements may be some boundaries at which individuals' strategies for action cleave. Thus, we might expect that at these boundaries groups of individuals are drawing on a shared set of resources, as well as on additional resources that are not shared equally by all others, to make interpretations of their surroundings that subsequently guide their action.

Figure 1 Framing Cultural Resources



Frames and Problem Construction

Frames are powerful means for shaping action because they guide both selection and salience. By directing attention, frames guide people to select certain aspects of the environment as important. Once selected, those aspects become salient—made more noticeable or meaningful—than others. Increased salience raises the likelihood that individuals will discern meaning from the object of focus and commit that meaning to memory (Fiske and Taylor 1991). Because frames act as templates for action, Collins and Pinch (1982) argue that members of a particular culture are likely to construct frames to help them deal with situations that are ambiguous and uncertain, and for which few routines and coping strategies exist.

Within organization studies, pioneering work by Orlikowski (1992, 2000) has shown researchers that interaction with a new technology marks an occasion in which much ambiguity and uncertainty exists, and for this reason, people will likely construct frames in these occasions to help guide their action. The specific concept of a "technological frame," arising from a paper by Orlikowski and Gash (1994, p. 178), refers to those instances in which people use cultural resources to make sense "not only of the nature and role of the technology itself, but the specific conditions, applications and consequences of that technology in particular contexts." The authors suggest, and others using the concept of the technological frame have shown (e.g., Davidson 2002, Edmondson et al. 2001), that different groups within the same broad organizational culture may construct radically different frames to interpret the technology; their departmental membership, occupational orientation, and technical knowledge base influence their selection of cultural resources. Because technologies are interpretively flexible—meaning is attributed to them as opposed to existing within them—members from different groups can indeed construct distinct frames that guide their interpretations of technology's functionality (Orlikowski 1992).

In the earliest phases of the technology development process—before there are detailed designs and long before the technology is built and implemented—innovators are not dealing with technological artifacts per se, but ideas about them. For this reason, the recognition that technologies themselves can be seen to have different meanings is less important or instrumental than the fact that people have different interpretations about the problems a technology is supposed to solve (Kaghan and Bowker 2001, MacKenzie 1996). Problem definition is not always a straightforward task because problems do not exist "out there" waiting to be found and solved. During its earliest stages, innovation might best be cast as a process of problem construction. Successful innovators examine a context and shape the "messiness" that they encounter into particular problems (Kaghan and Bowker 2001, p. 262). Indeed, as Carlson (1992, p. 175) indicates, "It is useful to think about inventors not as problem-solvers but instead as bundles of solutions who construct problems suited to their unique skills and ideas." This idea is similar to Swidler's (1986, p. 281) observations that people develop "strategies of action for which they have the cultural equipment." However, Swidler's (2001) own empirical work shows that people have an abundance of cultural resources available to them at any time, many of which they do not use. So, in the early stages of technology development, what draws frames around certain cultural resources as opposed to others, consequently leading innovators to construct one problem and not another?

Although this is certainly an empirical question, a rough framework for an answer to it might be found in work on organizational decision making. The influential work of Cohen et al. (1972) conceptualizes organizations as collections of solutions looking for problems. Problems are normally only defined as such if organizations can envision ways of solving them. Thus, the dialectical relationship that Cohen et al. identify is that solutions are constitutive features of problems: "Despite the dictum that you cannot find the answer until you have formulated the question well, you often do not know what the question is in organizational problem solving until you know the answer" (p. 3). In other words, an idea for a new technology may not only be a solution to a particular problem but also a part of the construction of that problem. Just as many decisions are bumbling about in the garbage can (Cohen et al.) waiting for choice opportunities, many cultural resources are available in an organization's toolkit waiting to be used to construct a problem. Although, in theory, any resource in the toolkit may be drawn upon to interpret a mess, it is likely that the actor will not know which resources are most appropriate. Conceptualizing a technology that could actually be built may allow the actor to concretize his understanding of what cultural resources he needs in order to mobilize his action. In other words, thinking of a technology's functionality helps to guide one's understanding of what can be done, and, with that understanding, knowledge of a workable solution to a problem becomes apparent. Conceptualizing technologies provides a way to turn a mess into a specific problem by ordering it in a way that renders it solvable.

Here, it seems useful to distinguish ideas about technology (hereafter, "technology concepts") from technological artifacts. A technology concept is an innovator's vision of what functionality the built technology (the technological artifact) should have. In Figure 1, then, the black and gray boxes represent broad technology concepts that draw frames around clusters of cultural resources. Those framed cultural resources direct the innovator's attention toward specific aspects of her environment, helping her to select aspects of messes that are important and to view them as salient. As she begins to focus her attention in particular ways, her understanding of a particular problem becomes sedimented, allowing her to deepen and refine her specification of the technology concept. In other words, technology concepts frame cultural resources, which then provide innovators with a manageable toolkit with which to construct problems. Constructed problems beg for solutions to legitimize them as problems, and innovators again pick up the tools in their cultural toolkit to help them make their broad technology concept specific. Therefore, during the development process, culture does not shape a technological artifact directly and unilaterally; rather, technology concepts frame cultural resources, which are then used to construct the very problems the technological artifact will be built to solve. Concordantly, a technology concept plays a fundamental role in selecting which cultural resources will be used to eventually develop the technological artifact.

If different groups use diverse technology concepts to frame cultural resources in the toolkit, they may consequently construct different problems. However, because the technology concepts are out there for discussion from the start, and once groups begin to discuss solutions they rarely revisit the problems they are supposed to solve (Kaplan 2008), it seems likely that members from different groups will disagree about what technological artifact should be developed without ever recognizing that they are using distinct frames to construct different problems. Debates then revolve around the more superficial issues of what features a technological artifact should have. Such debates may misdirect innovators, leading them to believe that they are simply disagreeing about the technological artifact and not about the more fundamental issue of what problem is most important to solve in order to achieve a common organizational goal. Here, we see the full implications of the fable of the six men and the elephant: Innovators are blind to the fact that others have constructed different problems than they have. Although they may recognize that others interpret a technological artifact differently than they do, they fail to see why they do and miss the opportunity to understand that they are actually in the midst of constructing very different problems and their identification of the technology concept is itself part of that process.

I explore this framework empirically through qualitative data on the development of a new technological artifact called CrashLab at a major U.S. automaker. The analysis shows how members of organizational departments became blind to the reasons why people in other departments did not share their understanding of what features CrashLab should have. The data also illustrate how this blindness can impede the innovation process unless organizations can find ways to help innovators move forward with development despite the fact that they do not understand the nature of their disagreement.

Site and Methods

Autoworks[1] is a large automobile manufacturer headquartered in the United States. Broadly, engineering efforts at Autoworks were divided into two specialties. The first specialty consisted of designers. Designers worked in a world of computer-aided drafting (CAD). They drafted their parts with CAD technologies and worked with vendors to provide material specifications. To make sure that their parts met a number of performance requirements so that ultimately the vehicle is sellable in the United States and foreign markets, designers work closely with engineers in a second specialty. These analysts were responsible for validating and testing the parts as they were assembled into complete vehicles. A designer would create her part with CAD software and assign it vehicle coordinates (on a three-dimensional grid) that indicated where that part would be located in the vehicle. Analysts assembled the CAD parts, based on these vehicle coordinates, into a fully integrated model that contained all the parts in a vehicle, as well as the connections between them. Analysts were then responsible for testing the performance of the complete vehicle on a number of different parameters, including crashworthiness, noise and vibration, aerodynamics, heat transfer, and cooling, among others. To complete these performance tests in a virtual environment, analysts used some form of finite element analysis. Finite element analysis is a computational technique that divides the actual geometry of a part into a large, finite collection of small, discrete triangular- or rectangular-shaped areas called finite elements. The elements are joined together by shared nodes. Nodes are the locations where values of unknowns (usually displacements) are approximated.

In this study, I focused on a technology conceived of to aid one class of analysts: crashworthiness engineers. Crashworthiness—the vehicle's ability to absorb energy through structural deformation—can be assessed either prospectively, using finite element models, or retrospectively by analyzing the results of crash test data. Crashworthiness analysts at Autoworks do both. To be legally sellable in the United States, each of the Autoworks vehicles had to meet basic crashworthiness requirements outlined by the National Highway Traffic Safety Administration (NHTSA). To meet—and typically exceed—these requirements, Autoworks used virtual vehicle assessment procedures to test over 120 different crashworthiness events. For a given vehicle, normally no less than 15 analysts worked together to evaluate a host of different crash testing conditions (called loadcases). In an attempt to reduce the time and effort it took analysts to set up and analyze a finite element model, as well as to standardize the assumptions engineers used throughout the process, innovators in Autoworks' research and development (R&D) department initiated a concept for new technology called

CrashLab. CrashLab was a computer simulation technology that that could be used for preprocessing (setting up simulations in ways that can be analyzed by a solver[2]) the finite element models that crashworthiness engineers use to make predictions about how a vehicle structure will respond in a given crash event and postprocessing (how the results obtained from the computational analysis can be extracted in such a way as to have predictive power) the models.

Data Collection and Analysis Procedures

The data presented in this paper were collected as part of a larger multimethod study aimed at understanding how organizational structure and technological artifacts coevolved across activities of development, implementation, and use (Leonardi 2007b; 2009, 2011). As I engaged in this larger study, I quickly learned that the development of the CrashLab concept was fraught with difficulties and required substantial compromise and organizational change to produce a working technological artifact. Unlike most contentious technology development processes that are aborted before a technological artifact is built, the development process at Autoworks resulted in a working artifact. Thus, CrashLab's development represents a "revelatory case" that allows organizational researchers to observe a process and its associated mechanisms that have not previously been observed or adequately described (Yin 1994). I focus here only on the development of the technology concept for CrashLab. Because the first technology concept for CrashLab was presented at an Autoworks R&D lab meeting in June of 1995, I knew that if I wanted to understand how the technological artifact was developed, I had to go back to some time before 1995 and follow it through to September of 2005 (the scheduled implementation date). Most of the activity associated with CrashLab's development had already occurred; thus, I would have to use ethnographic interviews as my primary source of data.

Interview Data. I began the interview process by identifying informants who I knew were involved in CrashLab's development, and through these interviews acquiring the names of others who were also involved. I interviewed members of four different departments within Autoworks: Safety, R&D, Information Services (Infoserv), and Technology Production (Techpro). I also conducted interviews in another department called Best Practice Evaluation (Bestpra) that was created to help develop CrashLab. Additionally, I conducted interviews with informants in three technology supplier organizations who helped, at various points in time, to build CrashLab: Link Engineering, Element Engineering, and Dynamic Engineering. Finally, I also conducted several interviews with other individuals who were not members of any of these organizations, such as Autoworks' senior management, consultants from other supplier firms, and professors at nearby universities who helped develop code for an early CrashLab prototype. Table 1 summarizes the different departments included in this study and key actors (who appear later in this paper) from each. Table 2 summarizes how many interviews were conducted within each organization and the number of informants who were interviewed. In total, I conducted 58 interviews with 36 informants about the development of CrashLab.

All interviews aimed to solicit from informants their recollection of events surrounding the development of

Table 1 Departments and Actors Involved in Development of CrashLab

Department	Function	Key actors (pseudonyms)
Safety	An engineering department staffed by analysts who use math-based simulation models to evaluate the crashworthiness of vehicle structures.	• Gene Anderson (Director) • Brad Bertallini • Sunil Kitane
Techpro	Responsible for assuring that all Autoworks engineers across the globe had access to, and training on, technologies for math-based analyses.	• Jensen Lu • Martin Yuen • Andre Brennan
Infoserv	Responsible for basic IT support to users of all technologies at Autoworks and for negotiating contracts and licenses with vendors.	• Kaleeb Azbad • Anthony Bernardo (Director)
R&D	Responsible for investigation into new processes and products for Autoworks' engineering and manufacturing operations, and new products for vehicles.	• Jim Beamer (Director) • Jason Chan
Bestpra	Responsible for producing standard work procedures for crashworthiness analysis.	• Arnold Dong • Rob Hickland
Focus Group	Responsible, among other things, for endorsing and sanctioning the use of all new technologies used by the analysts in a particular function. Staffed by representatives from related departments at Autoworks'.	• Sunil Kitane (Chair) • Gene Anderson • Jenson Lu • Kaleeb Azbad • Jason Chan
Link Engineering	External software vendor that produced the prototype of CrashLab.	
Element Engineering	External software vendor that built second version of CrashLab.	
Dynamic Engineering	External software vendor that build final version of CrashLab.	

Table 2 Summary of Interview and Archival Data

	R&D	Infoserv	Techpro	Safety	Bestpra	Link	Element	Dynamic	Other	Total
					Organization interviewed					
Number of interviews	12	6	11	14	5	2	1	2	5	58
Number of informants	7	4	6	9	3	2	1	2	5	36
					Archival documents collected by organization					
Number of pages	435	226	386	95	142	295	226	163	254	2,208
Number of documents	32	16	28	8	13	11	16	10	18	152

CrashLab, as well as their interpretations about why certain actions occurred. I began each interview by asking informants general questions about their employment experience at Autoworks, the role of their department in the vehicle development process, and the nature of their work. I then asked more specific questions about the development of CrashLab and changes that occurred in the organization of work at Autoworks over time. I also hoped to uncover more objective material, such as the composition of the teams involved in CrashLab's development, the fees paid to supplier organizations for their services, and the timing of certain events. To overcome information recall bias, I asked multiple participants these questions on multiple occasions (when they were interviewed more than once) to triangulate as best I could. In addition to gathering these more objective data, I was also interested in understanding the more subjective interpretations informants made about the events that unfolded during CrashLab's developments, their feelings about the direction in which CrashLab evolved, and their goals for the new technology.

Each interview was audio-recorded on a digital recorder with the informant's permission. Interviews lasted anywhere from 45 to 180 minutes in length. With the digital recorder running, I was free to take notes on a sheet of paper about items informants mentioned that were of interest, issues they raised that I had not considered, the names of further people with whom I should talk, and questions that remained unanswered. Before each interview, I reviewed the notes made during all previous interviews to determine what questions had still not been answered and about what issues I wanted to seek alternative interpretations. I then used this information to construct specific questions for the following interview. All recorded interviews were later transcribed verbatim and used as raw data for analysis.

Archival Data. During the interviews, many informants referenced documents, reports, e-mails, and presentations that had been generated during the course of CrashLab's development. At the end of each interview, I asked informants if I could borrow and copy the archival data to which they referred. In almost all cases, informants agreed. There were several times when informants could not locate the material they referenced. In these instances I followed up several days later with phone calls and e-mail to see if I could procure these documents. Upon reviewing this archival data (summarized in Table 2), more questions would surface about the development process, and I would incorporate these questions into subsequent interviews.

I collected more than 2,200 pages of documentation pertaining to CrashLab's development. This archival material consisted of 152 documents, including user specifications, formal reports, white papers, contracts with vendors, personal e-mails, memos, and presentation slides. These data evinced the rhetoric used by various groups in their formulation of objectives and cultural constructions of various problems CrashLab should solve. Moreover, they demonstrated how certain issues were brought to the fore, were contested, and then disappeared or were considered less important over time. I also gathered organization charts and briefing memos describing the structure and hierarchy at Autoworks at different points in time since 1995 so as to be able to match informant's accounts of organizational changes with formal alterations in Autoworks' structure of operations.

Analysis. I began analyzing the interview and archival data collected about CrashLab's development by treating each department's perspective as a separate case. Following the work of other innovation researchers, I took the approach that because each department sat in a structurally dissimilar position to all others within the Autoworks organizational structure (e.g., R&D versus Product Engineering), such that differential frames held by each department, and thus members' interpretations about the events that took place in the development of CrashLab, would be similar within departments and different between them. By first examining each department's account of CrashLab's development as a separate case, I avoided the tendency to gloss over distinctions that could prove important because they did not resonate in the narratives created by informants from another organization. By reviewing all of the interviews conducted within a given department, and by referencing the associated archival material, I wrote a running narrative of how all of the events unfolded from the perspective of that particular organization. I repeated this process for each of the organizations included in Table 2. These eight cases ranged from 15 to 60 pages in length. I used the cases for three purposes. First, I compared the facts depicted in them to triangulate the order of events involved in CrashLab's development; second, I determined what problem each group focused on; and third, I ascertained what technology concept each department advanced to solve that problem.

For the next step in the analysis I constructed a coding scheme to identify four cultural resources outlined by Swidler (2001, pp. 104–107): (1) values, (2) beliefs, (3) rituals, and (4) knowledge. After dividing all of the raw interview and archival data into groups corresponding to the four departments central to CrashLab's concept development—Safety, Techpro, Infoserv, and R&D—I used Atlas.TI©, a software program for qualitative analysis, to code passages from the data that evinced any of the four cultural resources. Following the practices outlined by Strauss and Corbin (1998), I compared the codes placed on data obtained from members within one group, and then on data between different groups. This process of comparison allowed me to determine which cultural resources were common within a department as well as which cultural resources were shared across departments and which were not.

Next, to understand the broad frame that members of each department brought to the problem construction process, I followed Bijker's (1995) suggestion to focus on four constitutive elements: (1) the goals members have for a technology concept, (2) the key problems they believe have to be solved to achieve the goal, (3) the strategies they use to solve those problems, (4) the requirements that have to be met for a solution to work. I used these four categories to again code the raw interview data of informants from the four departments. This process helped to reveal how each group constructed a particular problem and how that problem helped them to refine their broad technology concept. I then compared the constitutive elements of these frames within each group and across the groups to determine which were shared and which were germane to a particular group.

I then employed a "selective" scheme (Strauss and Corbin 1998) to uncover the steps departments took as they attempted to work together to move CrashLab from a technology concept to a technological artifact. At the highest level, I was looking for patterns that explained what steps constituted CrashLab's development process. At a more fine-grained level, I was interested in those specific activities that each organization performed to accomplish these steps. To connect these levels of analysis, I used Strauss and Corbin's (1998) method for "axial" coding, through which I first linked the codes at each level of analysis together by constructing subcategories that regrouped the data into clusters of similar activities. This iteration between data and concepts ended when I reached "theoretical saturation," or enough categories had been identified to explain how and why each department developed the problem they did and how discussions about these problems and/or the technological artifacts that were supposed to solve them affected the ongoing negotiations central to the innovation process.

Problem Construction

In this section, I demonstrate how each department used technology concepts to frame cultural resources that led them to construct distinct problems. Table 3 summarizes similarities and differences between the departments in terms of their technology concepts, frames, cultural resources, and constructed problems. As the table indicates, all four departments shared a common goal at a superordinate level, and also agreed on what activity they perceived that a new technology should substitute. However, each department developed distinct understandings of what the problem was, what strategies and what kind of knowledge had to be used to solve it, and, finally, what requirements had to be met so that one would know that a technological artifact successfully solved the constructed problem.

Safety: The Accuracy Problem

Gene Anderson, the director of Safety, had long felt that performance-engineering work did not occupy a central role in Autoworks' vehicle development process. Instead of providing proactive guidance to designers on how their parts should be designed, analysts reactively validated the performance of parts already created by them. Gene and the managers who worked beneath him were convinced of the value of increasing the effectiveness of the vehicle development process, and were certain that if they could be more proactive in their recommendations to designers, they would be able to contribute more meaningfully to it. For designers to act on their recommendations, however, the work of analysts had to be taken more seriously within the company. Gene and his managers reflected on their experience with the manufacturing department, who recently had adopted computer-aided manufacturing technologies and used them to increase the accuracy of tolerances throughout the vehicle. As a consequence of improving their accuracy, manufacturing had gained more leverage in decision making in cross-functional meetings. Barrett, one of Gene's managers, suggested that perhaps one way to have people outside of Safety take simulation work more seriously was to come up with a technology that would help to make simulation work more accurate.

As they looked for places in the analysis process that could be automated with the use of a technology, the Safety managers focused their attention on two ritualistic model-building practices: meshing and assembly. To translate CAD geometry into data that could be used for finite element analyses, analysts had to "mesh" CAD parts. A finer mesh produced a more accurate depiction of the deformation of a part because there was more data the solver could use to approximate displacements. A courser mesh did not capture the deformation of the parts as accurately and would provide results that would not correlate accurately with data from the physical crash. The

Table 3 Technology Concepts, Frames, Cultural Resources, and Constructed Problems at Autoworks

	Technology concept	Frame (constitutive elements)				Cultural resource (comprising toolkit)				Constructed problem
		Shared		Not shared		Shared	Not shared			
		Goals	Perceived substitution function	Problem-solving strategies	Requirement to be met by solution	Value	Belief	Ritual	Knowledge	
Safety	A technology that would generate uniform mesh and detect penetrations	Move simulation analysis earlier in the vehicle design process	Engineers working in an idiosyncratic manner	Change the procedure engineers use to build models	Engineers needed to build better mesh and reduce part penetrations	Increase effectiveness of organizational processes	Technology should automate routine processes	Conduct routine crashworthiness work	Product engineering, knowledge of vehicle crashworthiness dynamics	The accuracy of math models
Techpro	A technology that would fit into an already-envisioned flow of speed-enhancing technologies	Move simulation analysis earlier in the vehicle design process	Engineers working in an idiosyncratic manner	Develop a suite of interdependent technologies to achieve 2-day finite-element model analysis target	Math models had to be built so they could run faster, and be built in the same way in different parts of the world	Increase effectiveness of organizational processes	Technology should perform more accurately and quickly than engineers	Evaluate strength of computer-aided engineering and productionize new technologies	Understanding of engineering operations in global context and vendor relations	The speed of simulation analyses
Infoserv	A technology that would be more economical to maintain than current alternatives	Move simulation analysis earlier in the vehicle design process	Engineers working in an idiosyncratic manner	Consider options for combining technologies and increasing computing resources	The number of technologies for simulation work must be reduced. Size of models to be compliant with computer-processing capabilities	Increase effectiveness of organizational processes	Technology should have multiple features bundled into a uniform interface	Service and update all technologies across Autoworks	Contractual obligations with vendors and costs of solving finite element differential equations	The capacity of personnel and technical resources
R&D	A technology that allowed users minimal discretion in decision making	Move simulation analysis earlier in the vehicle design process	Engineers working in an idiosyncratic manner	Identify variance between simulation and test procedures	Engineers had to begin to pre- and postprocess their models in identical ways	Increase effectiveness of organizational processes	Technology should reduce variability in work practices	Investigate new methods for improving products and processes	Methodological orientation toward reproducibility and falsifiability of results	The credibility of simulation results

finer the mesh was, the longer it took the solver to generate the simulation. Because crashworthiness analysts shared supercomputing resources with other functions, computing capacity was at a premium. Safety managers determined that one way of reaching a balance between accuracy and processing time was to build models with varying densities. In other words, the part of the vehicle that was of most interest to the analyst would be meshed at a finer density than parts that were of less interest.

Model assembly was a second practice upon which crashworthiness managers focused their attention. When an analyst to simulated a vehicle's crashworthiness, she would enter into a shared parts database, find the CAD files for all of the parts that needed to be included in the model, and assemble them. During assembly, many things could go wrong. For example, designers drafted and updated parts at many different times. It was possible that a designer who was responsible for part A may have changed the shape of his part, whereas another designer responsible for part B had not. If the changes in the shape of part A made the part larger, it might then penetrate the space in which part B was located. Penetrations mattered immensely in crashworthiness analysis because they affected the solver's ability to predict the results of a particular test. For most analysts, spending a week attempting to resolve penetration issues was not acceptable, because doing so would often mean missing important deadlines. Analysts had to make a calculated choice about what penetrations they needed to fix in their models and which ones they could leave.

Convinced that the inaccuracy of simulations could be attributed, in large part, to the way analysts meshed and assembled their simulation models, Gene and his managers drew on their knowledge of crash dynamics to envision specific features of the technology that would help them rid Safety of this accuracy problem. The technology that they conceived would automatically create an finite element mesh that was configured (by way of varying densities) to balance mesh size and CPU processing time for the loadcase analyses analysts ran. Additionally, the technology would provide model checking capabilities to let engineers know if the most current version of a CAD file had been used to create the mesh, and point out penetrations that would affect the accuracy of the model. As Gene suggested, identifying a potential technology that could help crashworthiness analysts achieve better correlations between their simulations and physical tests was an essential step in showing to himself and to others that low levels of accuracy in finite element models were not just a matter of course, but a problem that could be solved.

Techpro: The Speed Problem

In 1995, crashworthiness engineers conducted nearly 650 crash tests—the most physical tests conducted at Autoworks in a one-year period. This milestone was not celebratory for innovators in Techpro who were trying to push the product-engineering departments (like Safety) away from physical testing and toward math analysis. To convince upper management that complete virtual vehicle assessment was valuable and that such a procedure could reduce the costs associated with vehicle testing, Jensen Lu and several other innovators in Techpro initiated a program that they called the "2-day challenge." The idea behind the 2-day challenge was that from the moment an analyst downloaded a file of a vehicle part it should take no more than two full days to complete a comprehensive crashworthiness analysis.

As members of the group responsible for the "global productionization" of simulation technologies, Techpro innovators also had to attend to how analysts at Autoworks centers in other countries were building and analyzing models. Lacking the manpower to perform all aspects of model building and analysis in house, the foreign centers often outsourced engineering analysis work to local supplier firms. Outsourcing was troublesome to Autoworks' senior managers because it increased the likelihood that proprietary information about vehicle design could be leaked to competitors and because it was costly. For management in Autoworks' foreign engineering centers, these two concerns were more than worth the risk when one considered the amount of time they saved by outsourcing model building.

With Autoworks' global engineering centers eager to outsource their work, Techpro innovators faced a clear challenge: they had to implement technologies that would make the model-building process just as fast in house as it was to outsource it, or analysts in Germany and Australia, for example, would continue to use supplier firms to build models. Martin Yuen, a manager in Techpro, underscored the importance of this point:

> If we compare [the way U.S. analysts work] to engineers in Germany, in Australia, they are doing things quite differently. For example, they don't do setup by themselves; instead, they just send it to the vendor—they have a local vendor that can do that for them—they give them, I think it starts from the [CAD] geometry. They give them the geometry and usually they pay for twenty or forty hours' worth of work.... Based on our experience with analysts [in Michigan], it takes much, much longer than the time the vendor takes. So, when we try to promote or implement simulation technologies in Germany, they have concerns; they have to know our technology is faster to justify not just paying money and sending the work to the vendor.

Because Techpro aimed to productionize and globally distribute simulation technologies that would reduce the need for Autoworks' product-engineering departments to leverage external resources to conduct math-based analyses, developing technology to make in-house model preparation faster than outsourcing was a primary concern.

Focusing on the practices analysts used to build models both domestically and abroad, members of Techpro quickly reached consensus that the substantial amount of time required to run math-based analyses was problematic. They believed that on the domestic front, the speed problem prevented math-based analysis from taking a more central role in the vehicle design process, and on the international front, the issue was costing Autoworks millions of dollars each year in outsourcing contracts. Working to fulfill the two-day finite element model goal, Techpro innovators generated a number of concepts for technologies that would help to speed up model building. Concepts for three interrelated technologies proved very promising. The first technology, rapid meshing assembly, would automatically translate part geometry into finite element data. After the geometry files were translated into mesh models, engineers could then employ a second technology known as material model checker. This technology was to be used to ensure that all of the correct material properties had been assigned to the parts. Upon completion of the material model checking, analysts would then employ a third technology known as vehicle model assembly, which would call all of the parts associated with a particular model into one central location so they could be arranged into a full-vehicle model.

Jensen Lu was confident that if analysts used these technologies in conjunction with one another they could reduce the time spent setting up models. As pleased as he was with the suite of technologies under development, Jensen still felt as if they were missing one final step in the process. To achieve a fully analyzable finite element model in the two-day period, analysts still had to instrument the model for the specific crash event they wished to evaluate by placing accelerometers on the model and positioning the model and the barrier in relationship to one another per NHTSA specifications.[3] This last process of model setup alone took analysts three or four days to complete. Thus, Techpro innovators still had to identify one final technology to complete the suite of technologies that would allow finite element models to be analyzable within two days. By mid-1995, Techpro innovators knew they had a hole in their technology lineup and that whatever technology they identified had to work in tandem with the rapid meshing assembly, material model checker, and vehicle model assembly technologies to increase the speed at which analysts could setup a finite element model.

Infoserv: The Capacity Problem

Autoworks' finite element computing environment in the mid-1990s was eclectic. With more than 20 different technologies to choose from when preprocessing, postprocessing, and solving simulation models, it was rare for two analysts to use the same technologies. By 1995, Infoserv found itself trying to clean up the mess that had been wrought by a highly decentralized technology-purchasing structure. Infoserv found it too costly to maintain licenses for so many different technologies and too laborious to provide updates and guarantee service for multiple technologies that performed the same task. One innovator in Infoserv recalled his astonishment at the diverse technologies used in product-engineering departments at Autoworks:

> [W]hen I came to Autoworks I was blown away. It was like whatever technology somebody wanted he got. I guess it evolved because all the different organizations operated as separate departments and didn't get really consolidated until the early 1990s. But still, the scene was crazy. There were way too many technologies and I told my boss one of the first things we have to do is consolidate and he agreed.

Autoworks' top management was pushing Infoserv to enable engineers to conduct more simulations so as to reduce the number of physical tests. However, relative to its user community, Infoserv was small. It did not have the workforce to purchase, service, maintain, and update whatever technologies analysts wanted to use. Further, the extent to which analysts could easily perform their analyses relied tremendously on Infoserv's ability to support the technologies on the front and back ends. Thus, Infoserv had to strike a balance between analysts' requests for flexibility to use the best technologies to do the analysis job and its ability to maintain those technologies. To simultaneously satisfy these two demands, Infoserv innovators established, in 1995, a plan geared toward reducing the number of technologies used in various product engineering departments, one of which was Safety.

As Infoserv began to consider how to reduce the number of technologies used for simulation analysis, it focused its sights on another capacity issue: the capacity of its supercomputing department to process the increasing number of simulations submitted by analysts. By the end of 1995, analysts in Safety were collectively running nearly 800 simulations per year on supercomputers. At this time, the average computing cost to run one simulation was nearly $20,000. This meant that crashworthiness engineers alone were responsible for more than $16 million in supercomputing costs per year. Although Infoserv innovators expected costs per simulation to decrease over time, they also anticipated that advances in technology, coupled with reductions in hardware testing, would beget an increase in the number of simulations conducted. A steady increase in the number of simulations being processed per year by supercomputers was only part of Infoserv's concern. Each year, the size of finite element models (measured in elements) had been increasing monotonically. By 1996 the average size model submitted to the supercomputing by Safety contained over 100,000 elements, a 400% increase in size from just eight years earlier. If by the year 2000 the average model contained 300,000 elements and the average number of models submitted to the supercomputing center in that year was 30,000, it was possible that the center would not have the capacity to solve all the models that were submitted.

In the minds of Infoserv innovators, the two capacity problems—the capacity of engineers to service so many technologies and the capacity of supercomputers to process so many simulations—were inherently intertwined. Although Infoserv technicians were looking to reduce the overall number of technologies in the finite element portfolio, they were certainly not against considering new technologies, especially if one new technology could replace two or three less efficient technologies currently in use. In fact, many Infoserv innovators within the finite element department were convinced that identifying a technology that could function both as pre- and postprocessor could solve the capacity problem. They informally divided up the work to focus their efforts on the technologies in specific performance engineering organizations. One of Infoserv's managers, Kaleeb Azbad, and his colleagues devoted their attention to reducing the technology portfolio in Safety, because they discovered that Gene Anderson and his managers had coordinated efforts to increase analysts' use of simulation models, and they began to grow concerned that they would not be able to solve the capacity problem fast enough to meet the growing computing demand.

R&D: The Credibility Problem

When Jim Beamer became the director of R&D, he sought to augment the organization's product-based focus with a process-based focus. To move his department in this new direction, Jim asked Jason Chan, a senior engineer in R&D who had been working on technologies related to crashworthiness, to spend a few months sitting and working with analysts in Safety, learning firsthand about the difficulties they were facing in the areas of airbag simulation. Jason spent the first two months of 1995 working on routine crashworthiness-engineering work with analysts. Trained in a Ph.D. program and employed by an industrial research center, Jason came to Safety with an orientation toward crashworthiness work as a scientific activity. This orientation was not shared by analysts who felt that it was far more important to make sure a model correlated with the results of a test than to spend time making sure the methods they used to correlate their models could be easily discerned by others. Although this different cultural orientation toward crashworthiness analysis work did not cause a rift between Jason and the analysts, it did focus Jason's attention on a different set of practices that could lead to more effective building and analysis of simulation models.

During his time at Safety, Jason noticed that the steps analysts took to preprocess their models (preparing to submit them to the solver) were highly idiosyncratic. After analysts meshed their parts and assembled those parts into a complete vehicle model, they had to take a number of important steps to instrument the model for a particular analysis. While instrumenting the model himself, and while watching analysts preprocess their models, Jason learned that NHTSA had very strict procedures indicating where they would place accelerometers and how the vehicle had to be positioned in relation to the barrier. As a result, analysts often placed accelerometers in locations other than those at which they would be placed in the NHTSA tests, and they often positioned the barrier incorrectly for the specific loadcase they aimed to evaluate. If an analyst instrumented a model in ways other than how it was instrumented at the proving grounds or in government tests, the correlation between simulations and tests would be poor.

Operating within his scientific frame of reference, Jason grew very concerned about these issues because he believed engineering managers could have little confidence that the data they were using to suggest expensive changes to vehicle design were credible because analysts used idiosyncratic processes to generate the results, and, as a consequence, managers had no way of knowing how those data were produced:

> I was thinking that a reputable laboratory has to have a standard; people have to respect their work.... There are still many people out there who say, "You trick the model to get the result. So what?" And so, simulation modeling has that kind of reputation. And so, I think it...it's like you send some part to a lab; if you know this lab tends to fake results just try to please their customer that would be awful, right? Then they won't be a credible laboratory. But you know if they're a good lab, if you can see their process, you know what happened to get the results. Our simulations, they have to be more credible like those of a good lab.

Thus, for Jason, the only way that simulation analyses could play a more central role in the vehicle design process was if managers viewed simulation results as credible data with which to make programmatic decisions.

After returning to R&D, Jason began to envision a technology that would guide analysts through preprocessing by showing them how to instrument a model. Over the next three months, Jason worked to develop a concept for a technology called CrashLab. The name was a bold statement that the technology should make simulation analysis as systematic and as credible as physical tests conducted by scientists in laboratories. Jason envisioned that CrashLab would work in conjunction with the preprocessors, postprocessors, and solvers already in use at Autoworks. An analyst would use an finite element preprocessor to read in electronic architecture files and mesh them. Analysts would also continue to use these preprocessing technologies to assemble the models, define contacts between the parts, and add and subtract spot welds. Once the model was completely meshed and assembled, the user would export the file and read it in to CrashLab. In CrashLab, analysts would instrument the vehicle and, after the model was ready, export the file and submit it to the solver at the supercomputing. Jason drafted an image of what he thought this interface would look like and prepared a slide for an upcoming R&D meeting to present his idea to his colleagues. His colleagues in R&D were very excited by the prospect of CrashLab and made suggestions for adding several minor features.

Blinded by the Technology

As the foregoing demonstrates, innovators in Safety, Techpro, Infoserv, and R&D all shared the goal of moving simulation analysis into a more central role in the vehicle development process. Although this goal was held in common across members in the four departments, the problems they believed had to be solved to achieve it were markedly different. The analysis above shows that innovators within each department actively worked to construct a problem that was, from their perspective, solvable. As employees of Autoworks, members in each of the departments shared a broad toolkit of cultural resources. For example, they were all "car buffs," they all valued Autoworks and its products as a company, they were committed to producing quality vehicles, they all understood (to a reasonable level) crashworthiness dynamics, and they each had knowledge of finite element modeling and the skills with which to do it. As members of different occupational communities who came to their tasks with differing orientations and objectives (Van Maanen and Barley 1984), members of each department also had certain cultural resources from which to draw that others did not. For example, analysts in Safety had skills to recognize which parts of a vehicle could have more granular meshes than others and still produce an accurate output; members of Techpro had knowledge about how to actually build technologies to be interoperable, and they had knowledge of how engineers at Autoworks' global centers worked; members of Infoserv had authority to implement a technology reduction strategy and a broad understanding of the limits of supercomputer processing power; whereas engineers in R&D had norms taken from academic research programs for consistency and transparency in methods. As the analysis above demonstrates, some number of those cultural resources that they held in common with other departments and some of those cultural resources germane to their own departments were framed together into resources that were used to construct a specific problem. The technology concepts that each department came up with not only helped members determine which cultural resources to draw on in constructing the problem, but the very identification of the technology concept indicated a solution, which thereby rendered the problem valid. Consequently, each department had arrived at an understanding of (1) what problem needed to be solved to move simulation analysis to a more central role in the vehicle development process and (2) what the technology should look like that would solve that problem.

Emboldened by the positive feedback he received from his colleagues in R&D, Jason decided to present his proposal for CrashLab to the focus group. The focus group was responsible, among other things, for endorsing and sanctioning all new technologies used by the analysts in Safety. If the focus group did not deem a new technology sufficiently useful or appropriate, it would refuse to endorse it, and, ultimately, Infoserv could not install the software on analysts' machines. Sunil Kitane, a senior manager in Safety, chaired the focus group. Sunil had worked as a analyst in Safety for more than 15 years by the time the focus group was founded in the early 1990s. One of Sunil's early jobs as chair was to determine which engineers across the various departments at Autoworks were interested in and involved with product engineering from a crashworthiness standpoint, and ensure that these key players became members of the focus group. The focus group included, among others, Gene Anderson from Safety, Jensen Lu from Techpro, Kaleeb Azbad from Infoserv, and Jason Chan from R&D.

In June of 1995, Jason discussed his concept for CrashLab, providing each of the participants of the focus group meeting with a handout indicating what features CrashLab would have. This summary from the archived meeting minutes shows that Jason's discussion about CrashLab linked those features to the goal of moving simulation analysis into a central design role:

> Jason Chan presents concept for "CrashLab" technology and provides handout with proposed features (appended to minutes). Discusses how technology will automate selection of best practices and indicate extent to which user violates best-practice guidelines. Sunil Kitane asks why CrashLab is needed. Jason Chan indicates that it will help to make simulation work more predictive. Gene Anderson asks why it does not have model check features. Jason Chan indicates that model check features are not necessary if best practices are followed. Debate about features continues for several minutes.

Later, Jason recalled his frustration with the presentation:

> Everyone kept saying, "Why doesn't it have this feature and that feature." They all wanted different things in it. They did not understand its vision I guess. I told them why those features were not necessary but they still wanted them.

Gene Anderson's perspective was somewhat different:

> Jason's first presentation is very clear in my mind. It was like, "I thought we all had the same goal here. Why do we need some automated script to test for variance in practice?" It was flawed because it didn't have some important features that were essential to make math work more accurate like weld checking modules.

Jason's presentation was not met with enthusiastic support. Although the other members of the focus group agreed with Jason's assessment that Safety needed to move toward a completely virtual vehicle assessment procedure, each member felt that the technology concept was deficient in its ability to solve the problems that member believed were important. Because members of Safety, Techpro, and Infoserv had all developed their individual problems that had to be solved to meaningfully integrate simulation analyses into the vehicle development process, they expected CrashLab to have different features than Jason proposed. Consider the comments from focus group members:

> **Sunil:** I remember the CrashLab seemed like a good technology, but it wasn't ready for prime time. It did some important things like provide set filtering algorithms, but that wasn't really the major problem. We needed a technology that would make the results our engineers gave more useful, more real, you know, more accurate. The early version of CrashLab didn't do that. It was like it ignored that problem altogether. Our needs were more timely than that.
>
> **Jensen:** The CrashLab had some issues because there was no immediate way to make the work faster it didn't seem. We had all agreed to make the technology or the finding of a technology to work with the 2-day finite element model process. The CrashLab was good in spirit, but not in practice because many of the things it could do would make work slower.
>
> **Kaleeb:** When Jason explained the architecture of CrashLab I thought there was a problem because if [analysts] used it they would generate more models with larger file sizes and they'd quickly exceed their dedicated space on the R-drive.[4] Also, it was another technology, and more technologies we were thinking were bad since we were trying to reduce our numbers anyway.

Each member of the focus group was expecting CrashLab to be the technology that his department had identified to solve its particular problem.

The discussion about what features CrashLab should have continued for six months. As Kaleeb commented, "It was a brutal time. We were always battling it out over who wanted what features and no one could come to agreement. I just felt like everyone else was barking up the wrong tree. I didn't get why they didn't agree with me. It seemed so obvious." Kaleeb's comments indicate that innovators from one department were blind to the reasons why innovators from other departments wanted different features for CrashLab. More specifically, none recognized that each department had constructed a different problem. Because the conceptualization of the technology was an integral part of the problem construction process, a proposed change in features threatened to undermine the validity of the problem that department had constructed. The recognition that each department had constructed different problems was further obscured by the fact that all the members of the focus group shared a common goal: to move simulation analysis to a more central role in the vehicle design process. As Jensen noted,

> Sometimes someone would want some ridiculous feature that would be real expensive like the ability to automatically correct penetrations and I would have to ask if we still shared the same goal of making math analyses indispensible in the process. And everyone would agree so then you were left to assume that they didn't really get exactly what features were feasible or they couldn't link the goals to the operation of the tool.

It came as no surprise, then, that in its last monthly meeting of 1995, the focus group voted to not approve Jason's concept for CrashLab. Jason and his team in R&D were frustrated with and confused by the decision. As one of Jason's team members recalled, "We all wanted to build a technology to make simulation work more central to design, but everyone looked at CrashLab and saw different features they wanted. We just couldn't get why they didn't see the features that we thought were so obvious."

Discouraged from his experience with the focus group, Jason Chan began to concentrate on other projects, reducing his effort spent on CrashLab. Although Jason still believed that CrashLab could solve the credibility problem in simulation analysis, without support from the focus group there would be little chance that the technology would ever be used in Safety. Jason's team was convinced that the focus group had rejected CrashLab because they did not believe that it could be effectively built. Unaware of the deep-rooted concerns of the other departments represented in the Focus Group, Jason tapped into some discretionary money he had in his R&D account and continued slowly with the development of a working prototype, convinced that a proof of concept would remove any doubt about CrashLab's potential to revolutionize simulation analyses at Autoworks.

By 1998 the work of Jason's team finally resulted in a fully functional prototype of a virtual bumper module for CrashLab. The engineers in R&D were quite enthusiastic about the result and felt that they had succeeded in building a simple, intuitive technology that would increase the credibility of simulation analysis. Emboldened by the support he received from R&D, Jason pitched the idea for CrashLab for a second time to the crash focus group in the spring of 1998. His presentation was quite similar to the one he had made almost three years earlier, but this time at the end of his talk Jason pulled out a secret weapon: A working prototype of the CrashLab concept. Unfortunately, the focus group members remained unimpressed. CrashLab still did not resonate with the problems they had constructed in each of their respective organizations. Thus, the prototype alone was insufficient, in Star's (1989) terms, as a boundary object: It did not translate R&D's desire to solve the credibility problem into a logic that could be understood from within the frames held by the other departments.

So, much to Jason's dismay, the focus group again voted not to endorse CrashLab for production and distribution in Safety.

Despite the fact that CrashLab promised a number of ways to overcome the separate problems developed by the focus group members, individuals from the three other departments did not see the utility of CrashLab to solve their independent problems. As Sunil Kitane commented,

> I wasn't alone, you know, not being sure of what CrashLab had to offer. I mean I saw the prototype and it looked good, but for some reason it didn't resonate with me that this could make the work more accurate. I think part of the problem was that maybe it was how Jason pitched it or something, it just didn't click, you know?

So, why didn't it "click"? Jason was savvy enough to know that he had to pitch his idea to the group in a way that would allow them to see the advantages of the technology. What Jason had been unable to do successfully was to articulate his vision of CrashLab in ways that resonated within frames other than his own. That is, without positioning the perceived usefulness of CrashLab inside the frames held by other departments, he was unable to help the focus group members overcome the culturally constructed barriers that conceptually separated them and promoted their differentiation. Analysis of the archival documents showed that members from Safety, Techpro, Infoserv, and R&D engaged in three months of debate after this April 1998 meeting about the features CrashLab should have. The following excerpt from an e-mail from Andre Brennan, an engineer in Techpro, to the focus group listserv summarizes everyone's continued focus on features as opposed to problems:

> We've been discussing this for almost three years. It is essential for us [Techpro] that CrashLab integrate with the [Material Model Checker] technology. Jason seems insistent that such interoperability is not necessary and Gene continues to say that for the cost a penetration checker is more important. If CrashLab is going to help us be more proactive in simulation work it needs to integrate with the Material Checker. We have got to stop proposing all these extra features that don't serve clear purposes and focus on the ones that matter. Can we please try in our next meeting to agree in a final list of features?

Thus, it appears that the working prototype of the technological artifact intensified conversation and debate over which features CrashLab should have and, in so doing, further exacerbated the blindness that members had to the problems constructed by other departments. Jason had hoped that a working prototype with actual features would help to guide conversation and focus attention. In the literature on knowledge sharing, authors such as Carlile (2004) and Bechky (2003) would agree with him. Their studies show that artifacts that make the abstract concrete often help members from different occupational communities to translate or transform knowledge, as opposed to simply transferring it across organizational boundaries where it becomes lost in a sea of semantic incompatibility. In this case, however, making abstract features concrete only compelled members of the four departments to debate more about *what* the technology should look like as opposed to engaging in discussions about *why* the technology should have certain features. Three years after Jason first proposed the idea of CrashLab, innovators were still blind to the fact that people in other departments were hoping the technology would solve a different problem. Because of this blindness, the focus group voted again in the summer of 1998 not to endorse CrashLab for development and deployment at Autoworks.

Reintroducing Ambiguity and Reformulating Boundaries

Just like the six blind men who argued about the attributes of an elephant because they could not see that they were touching different parts, so innovators from the four departments argued over CrashLab's features because they could not see that they were all focusing on different problems. Because the blind men could not see the true nature of their disagreement, they were never able to truly know the elephant. Similarly, because innovators at Autoworks could not see that one's insistence on having particular features included in the technology was linked to the construction and definition of a particular problem that had to be solved before simulation technologies could play a more central role in vehicle design, they could not build CrashLab.

When ideas about a new technology initially emerged within each department, they activated the department's culture by clustering together resources that guided problem construction and selection of a technology concept. In other words, the technologies themselves were instrumental in selecting cultural resources from the broad toolkit that would be eventually be used by the department's members to advocate which features it should have. Through the process of constructing problems that the technology could solve, cultural differences between departments became more salient. These differences were noticeable to innovators at Autoworks. As Frank, an engineer in Techpro, commented, "I don't remember us [Techpro] being so different from Infoserv before we started talking about CrashLab. After it seemed like we're on totally different spectrums." Jason from R&D made a similar observation about his own actions:

> Sometimes I would sit in the meetings for CrashLab and get real mad that no one cared about scientific rigor. But then I was like, to myself, "since when do you care so

> much about that?" and I would think that I hadn't always been this way but for some reason I really cared when it came to the simulation work and CrashLab and that whole aspect of scientific culture we have in R&D was important. I can't explain it but it was just more intense around CrashLab.

In short, technology concepts framed a collection of cultural resources. That collection of resources acted as a frame through which innovators could interpret what problem needed to be solved. As the problem became more refined, ideas about what features the technology needed became more refined too. As members from other departments argued against those features, they were not simply challenging the innovators' vision of a technology, nor were the simply challenging their construction of the problem; they were also challenging the cultural action emerging out of the combination of resources framed by those innovators. That cultural identities would be invoked during periods of ambiguity and strengthen during disagreements with members from other departments is not surprising. As Swidler (1986, p. 279) notes, "When competing ways of organizing action are developing or contending for dominance people formulate, flesh out, and put into practice new habits of action. In such situations, culture may indeed be said to directly shape action." Thus, disagreement over what features CrashLab should have can be seen as clashes over cultural identities within departments, which helps to explain why innovators grew disturbed by the apparent flippancy of their focus group colleagues and became more steadfast in support of their own view of CrashLab's functionality.

By all accounts, CrashLab was officially "dead" in the fall of 1998 and the focus group was no longer discussing CrashLab at its meetings. But in December of that year, Anthony Bernardo came to Autoworks from another major automobile company to be the new director of the Infoserv department. During one of his early meetings with Kaleeb, he learned about R&D's prototype for CrashLab. Anthony was intrigued by the concept and scheduled a meeting with Jason to have a look at the technology. As a new member to Infoserv, and to Autoworks in general, Anthony was not deeply inscribed in a particular cultural frame. What Anthony saw in CrashLab was not necessarily a technology that could fix a specific problem (because as a newcomer to Autoworks he did not yet have a handle on exactly what problems people believed existed), but a technology that could fulfill a larger need in the broader world of computer-aided engineering: standardization. As Anthony commented, CrashLab held the promise of standardizing the work of disparate engineers through its capabilities for automation:

> [CrashLab] . . . is a way of capturing the standard. I mean, you know, if you look, go back in engineering organizations, the way they used to do it was the standards were in the standards book and they were on a shelf someplace; and so, you know, you can follow the standards. You have to go look them up, you have a complicated process. You're running maybe 15, 20 different safety loadcases. You want to look it up or you just want to have the process basically in front of you. So, CrashLab could really help get standard work in front of the engineer on a daily basis.

At the first focus group meeting of 1999, Anthony revived the idea of CrashLab. He mentioned that he believed CrashLab could help to "standardize" tasks. The use of the term "standardization" was quite powerful; a power that surfaced in large part because of the term's ambiguity. From one perspective, standardization can be seen to increase uniformity (cars drive on the same side of the road). From another it can be seen to increase speed (one does not have to look around for alternatives). From yet another perspective, standardization can be seen to reduce waste (common-sized nuts and bolts save industries millions of dollars each year). In this sense, the rhetoric of standardization was ambiguous: two people may well agree that standardization is a desired goal and even necessary, while holding very different ideas about what standardization actually means.

No matter whether Anthony's rhetoric of standardization was a consequence of planning or fortuitousness, it resonated simultaneously with people from Safety, Techpro, Infoserv, and R&D. When the focus group was dealing with Jason's prototype, their discussion centered on the features Jason had built into the technology. As one focus group member commented, "The features were there and staring us in the face so they became the center of discussion." Anthony advocated that the group leave Jason's prototype aside in their discussions and focus instead on what features they believed the technology should have if it was going to promote standardization. Because Anthony's positioning of the idea of standardization was so vague, it allowed each of the departments involved with the focus group to couch their specific problem under its umbrella. Consider the following definitions of standardization proffered by focus group members:

> **Sunil:** When work is standardized that means it is done more accurately like the physical tests.
>
> **Jensen:** Standardization . . . is so important because it leads to consistency and speed.
>
> **Kaleeb:** Standardization means that you don't have so many technologies to contend with.

The ambiguity of the standardization idea allowed members from the four departments for the first time to move away from discussion about CrashLab's "features" and to talk about a "problem" that a technology should solve. The "standardization problem" was now defined as a problem common to multiple organizations at Autoworks. However, no one ever sat down to ask exactly what standardization meant. Instead, each group interpreted standardization in light of the particular problem it had developed and believed that when other organizations were talking about standardization, they were addressing the same problem. As Sunil commented, "Now when I hear people say 'standardization' they finally get that we need to be more accurate in our testing." Of course, Sunil's assessment of uniform understanding in the group was incorrect. But Sunil, like the others, did not know that the group was not in agreement, and thus moved forward. Had any one member of the focus group probed deep enough to uncover what his colleagues meant when they talked about CrashLab fixing the "standardization problem" he would have quickly discovered that there was much less uniformity of opinion than previously thought. But such probing did not occur and after a few weeks of e-mail discussion following the meeting, the focus group was abuzz with excitement about the potential of CrashLab to solve the problem of standardization.

By the winter, the focus group gave Jason the green light to develop a second module for CrashLab: a virtual front-impact test laboratory to prepare and analyze a model per NHTSA standards. Although the focus group had still not officially endorsed the project, Jason was able to secure additional funds from R&D and Infoserv to continue exploration related to the development of the new module. Under the logic of standardization that now permeated the discourse of the focus group, CrashLab held great potential. Still, however, the focus group was not ready to officially endorse the project, because as Sunil pointed out, there was one crucial link missing: before CrashLab could standardize the work, standards had to be created: "The problem that I and some others had with CrashLab was that, I mean, at that time, we didn't even have anything standardized. We didn't have standard practices."

In early 1999, a senior manager at Autoworks reorganized boundaries in the engineering division to create a new department called Bestpra. The goal of Bestpra was to create and update best practices for crashworthiness analysis within the company. Structurally, Bestpra sat outside of regular product engineering. Unlike the focus group's cross-functional structure, Bestpra innovators worked only in their department. However, the kind of work that Bestpra performed cut across all the other functions. As Arnold Dong, a subject matter expert in the new department, indicated, "Bestpra would never had been organized as a department if it wasn't for CrashLab. CrashLab needed standardized work and so Autoworks needed to create a department that could come up with standards."

The six engineers who formed Bestpra were given the title of subject matter experts. Their role was to collect information from the analysts in Safety to determine what the standards (or standard work as it was often called) would be for a particular type of analysis. In March of 1999, the Bestpra organization published its first standard work guidelines for crashworthiness analysis. Jason and his group used the best practices identified in this document to determine how to automate the process of setting up and analyzing simulation models in CrashLab. Working with the engineers at a software supplier named Link Engineering, R&D implemented this new knowledge in the virtual frontal-impact test module. By the end of 1999, the focus group was very happy with the progress that the CrashLab technology was making. Now that all focus group members were enthusiastic about CrashLab and they all saw it as a viable technology for production, they finally endorsed it.

Now that Bestpra was responsible for standard work, and because CrashLab was commonly seen as a technology to enable "standardization," the new department held the locus of decision-making authority in regards to CrashLab's features. However, subject matter experts in Bestpra still needed input and buy-in from members of Safety, Techpro, Infoserv, and especially from R&D (because they had the most knowledge relevant to the technology) to be able to build those features. Thus, it was important for them to maintain ambiguity around "standardization" so that each of the departments could move forward even though they remained blind to their differences.

For this reason, many of the specific decisions made by Bestpra's subject matter experts with regards to CrashLab's functionality did not surface at the focus group meetings. They knew well that if they admitted to building more modules into CrashLab before refining and augmenting the algorithms for the existing modules, Jason and the engineers in R&D, among others, would be unhappy. Thus, as Rob Hickland, another subject matter expert suggested, the updates that Bestpra would present to the focus group seemed intentionally ambiguous:

> We were working closely with the people in Jensen Lu's group to implement more loadcases in CrashLab, but that seemed to be kept quiet. I mean at the Focus Group meetings Techpro would talk about working to develop new algorithms, but would sort of leave it at that. I think a lot of people knew that we were aggressively working to get the standard work for [other analyses] into CrashLab too, but a lot of people didn't know it.

A second area in which Bestpra was strategically ambiguous (Eisenberg 1984) about their work was in regards to the amount of discretion that an engineer should be allowed to have when using CrashLab to build and analyze a model. Although Jason had originally conceived of automation as a way to ensure that each engineer who used CrashLab was following common practices, he was aware that there were certainly instances in which it was in the best interest not only of the concerned engineer, but of Autoworks in general, for a analyst to veer from best practice. This decision to allow analysts to subvert the automated routines embedded in the software was meant as a signal that discretion was an asset rather than a liability and that R&D felt that engineering intuition and knowledge were of the utmost importance in good mathematical simulations. Engineers in Bestpra did not share the same view on the place of discretion in model building. As one Bestpra engineer commented, allowing the engineer to exercise her discretion to deviate from a best practice undermined the goal of best practices and discounted the important role his organization played in developing the standards in the first place.

As the findings suggest, the mechanisms of reframing the problem ambiguously as one of standardization and reorganizing the boundaries of the development process so that innovators from departments less central to CrashLab's ongoing production did not have a chance to connect the ambiguous representation of the problem with the concrete specification worked to move the development process forward. Table 4 details the features that each department envisioned that CrashLab would have, the conflict that focusing on these idealized features produced between departments, and how the two mechanisms allowed members of each department to live with their innovation blindness. However, although reintroducing ambiguity and reformulating boundaries allowed the development process to move forward so that CrashLab could actually be built, the table indicates that the actual production artifact did not represent the initial desires of each the department equitably. By comparing the asterisks from the far left column to the far right shows that Techpro and Infoserv (the departments who continued to be involved after the boundaries of the development process were redrawn) were disproportionately rewarded in terms of the number of their desired features that found material form in the production artifact. Thus, it appears that although living with innovation blindness might be essential to produce a working technology, the very mechanisms that spur the process along are effective in doing so precisely because they enable most innovators to remain suspended in a web of ambiguity, and only a minority ever leave that web to translate the ambiguous terms used to produce temporary agreement into actual technological features.

By 2002, the early development work and prototype design for CrashLab was complete. The actual coding and building of the technology would be outsourced to two software vendors, Element Engineering and Dynamic Engineering. Techpro and Infoserv, with the help of subject matter experts in Bestpra, remained responsible for CrashLab's development, with minimal input from Safety and R&D. The contract with Link Engineering specified that by April 2004, the software had to be ready for distribution and contain modules for a number of specific crashworthiness loadcases. After a number of delays, Link Engineering finally submitted a complete version of CrashLab ready to be installed on crashworthiness analyst machines in September of 2005. My research after implementation suggests that CrashLab was differentially received by two product engineering groups within Safety. Specifically, one group rejected the technology because they perceived that its functionality did not meet their needs (Leonardi 2009), whereas another group used the technology and dramatically changed the way they worked (Leonardi 2011). The differences between these two groups can be traced directly to the development dynamics outlined in this paper and the materiality of the technological artifact those dynamics produced.

Discussion

A burgeoning literature on technology development implicitly suggests that culture shapes the way that different groups conceive of new technologies. An important assumption underlying this suggestion is that culture frames what technologies people see. This study advocates an alternative, though complementary, starting point. By treating culture as a toolkit of resources that people can differentially draw on to create strategies for action (Swidler 1986), I have suggested that new technology concepts frame cultural resources, which are then used to construct the very problems the technological artifact will be built to solve. From this perspective, culture does not directly shape technology. Rather, technology concepts activate culture as they draw frames around resources that will guide people's problem construction practices. By acting as a frame through which problems can be constructed, technology concepts play a key role in selecting the set of cultural resources that will be used to develop the subsequent artifacts.

The case study makes clear that innovators from different departments disagreed about what features the technological artifact should have. More importantly, however, the findings show that innovators were blind to the reasons why others disagreed with them. As discussions became more focused on the specifics of the technological artifact, innovators moved further away from being able to recognize that the true nature of their disagreement was about the problems departments wanted to solve. Innovators never overcame their innovation blindness, but by reintroducing ambiguity into a process that had become relatively concrete, and by reorganizing boundaries in ways that provided a structural context in which ambiguity could succeed, they were eventually able to produce a working technology. Below, I discuss implications of these findings for theories about the relationship between technology and culture and for theories of collaboration in the innovation process.

Table 4 Mechanisms for Turning Technology Concept into Technological Artifact

	Envisioned technological artifact (features deemed most desirable by each department to solve constructed problem)	Conflict	Mechanisms for living productively with innovation blindness		Living with innovation blindness	Implemented technological artifact (actual features included in production version)
			Reintroduce ambiguity	Reorganize boundaries		
Safety	—Mesh generation —Part locator —Coordinate regeneration —Contact creator*	Fourteen features deemed "most desirable" by each department. None shared across departments. Innovators fight about the features while blind to the fact that features have been chosen as solutions to differently constructed problems	Standardization = Less room for error, more accuracy	Less visibility into development process after creation of Bestpra	The term "standardization" allows innovators from each department to speak in a common language about the problems they are trying to solve, even though those problems are different. This coordination allows them to agree to build CrashLab. Reorganizing the boundaries of the development process to include some departments more centrally and others more peripherally prevents paralyzing debate about what features will be included in the built technological artifact enough to turn CrashLab from a technology concept to a technological artifact.	—Material model checker —Section creator —Automatic model instrumentation (partial) —Automatic energy group formation —Automatic vehicle positioning —Automatic node identification —Initial condition script —Contact creator —Automatic occupant positioning —Automatic seatbelt router —Input deck writer —Report generation (without best practice violation alert)
Techpro	—Automatic vehicle* positioning —Automatic model instrumentation* —Automatic spot welding —Automatic occupant* positioning		Standardization = Faster model building and analysis time	More visibility into development process after creation of Bestpra		
Infoserv	—Mesh curvature formation —Material model checker* —Input deck writer* —Data compression		Standardization = Similar technologies, smaller number of programs	Same visibility into development process after creation of Bestpra		
R&D	—Flow chart —Best practice violation alert —Uniform data extraction modules		Standardization = More consistent, reliable results	Less visibility into development process after creation of Bestpra		

*Denotes features that were eventually included in the production version of CrashLab.

Contributions for Theories of Technology and Culture

Over the past two decades, students of organizational culture have seemed unclear about what to do with the concept of technology. Most major writings on organizational culture skirt around meaningful discussions of technology (e.g., Alvesson 2002, Martin 2002, Schein 2004). Alvesson, Martin, and others argue, for example, that technologies provide imagery (e.g., being high tech or fast moving) that is useful in storytelling and other practices through which culture is transmitted and stored. Schein (2004, p. 36) suggests that "though the essence of a group's culture is its pattern of shared, basic taken-for-granted assumptions, the culture will manifest itself at the level of observable artifacts." In this respect culture is constituted by activities that occur independent of technology and later get inscribed in the technology as frozen organizational discourse (Bowker and Star 1994). Presumably, over time, members who use those technologies are influenced by their constraints and affordances; consequently, the technology shapes cultural practices (Leonardi 2007a). Thus, although most authors take the relationship between culture and technology far enough to acknowledge that they are mutually influential, they continue to treat them as empirically distinct phenomena (see, for a discussion, Leonardi and Barley 2008).

The approach advocated in this paper suggests that in the early stages of the innovation process, culture and technology may not be empirically distinguishable. Technology concepts serve as frames for organizing cultural resources into interpretive schemes that can influence action. This treatment of technology as a frame is unique from discussions of "technological frames" (Davidson 2006, Orlikowski and Gash 1994). In the "technology as a frame" approach demonstrated in this paper, technology concepts are primary means by which cultural resources cohere into strategies for action. Culture is activated—made manifest in action—when technology concepts invoke the need for innovators to call on their cultural resources to construct a specific problem. As innovators draw on their cultural resources to refine problems, their conceptualizations of the technology shift so as to be capable of solving the changing problem. Thus, technology concepts may be used to select the cultural resources that influence the technological artifacts that are eventually built. In this approach, technology and culture are indistinguishable. As we saw in the case at Autoworks, each department's initial conceptualization of a technology activated a set of cultural resources such that members found that when they discussed CrashLab with others they began to more strongly identify with a departmental culture. Over time, the technology concept became a proxy for a department's culture such that debates about technological artifacts became debates about the cultural values placed on problems in the innovation process. By contrast, the "technological frame" approach suggests that cultural resources are used to interpret a technology. Thus, some other external stimulus activates cultural resources that actors will use to determine the uses to which the technological artifact can and should be put. What this stimulus is remains unclear or is inconsistent in writings on technological frames. In that approach, culture influences what people think of the technological artifact, and sometimes how they will modify or change its features. Although this approach may be adequate for studies of implementation, it may not truly capture the interplay between technology and culture during development. This study has shown that during development activities technology concepts are inherently intertwined with cultural resources such that the two act together to produce interpretations of what technological artifact should be built. Thus, what is normally referred to as a "technological frame" in studies of implementation might be better conceived of as a "cultural frame about technology" in studies of development.

The important distinction between these two approaches, then, is the phenomenon that stimulates the practice of framing. This study has suggested that in the earliest phases of the development process, technology concepts act as such a stimulus. At Autoworks, different technology concepts eventually became a singular technological artifact. However, one could imagine that in other contexts of innovation technologies could take other forms, such as routines, policies, or tasks. If, as Entman (1993, p. 53) suggests, culture should be considered as "the stock of commonly invoked frames" and that "culture might be defined as the empirically demonstrable set of common frames exhibited in the discourse and thinking of people in a social grouping," technology concepts that act as framing devices will be core to any production of cultural action. As such, the perspective advocated in this paper accords technology a central role in the activities that we call organizational culture.

The findings portrayed in Table 3 suggest that technology concepts trigger frames, which activate toolkits and imply problems. An adherent of the view of organizations as consisting of coherent subcultures might suggest that the "selection" or construction of technology concepts and subsequent problems is simply a consequence of cultural biases. Or, proponents might suggest that each subculture contains a unique repertoire of cultural tools. The findings of this study push back on this cultural determinism in favor of an interactionist view. Although certain cultural tools are unique to organizational departments, others are held in common. It is the differential combination of these tools into specific toolkits that guides action. I have shown that in product development, an organization's technology concepts may frame those tools into specific toolkits. In other organizations, concepts about policies, strategies, or other pertinent phenomena may act as the frames that activate culture. Such an approach helps to explain why it is that sometimes groups from different organizational departments walk in common steps and other times find themselves in bitter disagreements.

On a related note, when people interact with a technology, whether by developing concepts for it or physically modifying it, they always do so within the context of organizing. The process of organizing is accomplished through action and interaction. When a new technology becomes the focus of people's actions, whether those actions are of development or use, people's interactions will necessarily reflect this influence. However, these ongoing interactions constitutive of organizational culture continually shape the kinds of actions people perform in the context of a technology. For this reason, I suggest that technology and organizational culture cannot be viewed as distinct empirical phenomena. To uncover what consequences taken-for-granted assumptions about technological change can have on cultural forms of organizing, one must look at the relationship between social and material changes occurring throughout the concept development process. Changes in a technology's material features (e.g., building directories to house standard work practices) often necessitate social changes (e.g., the creation of a new Bestpra organization), which then catalyze more material changes. The main implication of this insight is that theorists of organizational culture must find a place for the role of technology as they expand their constitutive approaches. When organizational culture is viewed from a constitutive perspective, technology cannot be treated as an exogenous variable, but should be conceptualized as a practice that intertwines itself with other work and communication practices to constitute culture. Culture is tightly bound to the material characteristics of those core technologies that organizations develop, distribute, or service. Currently, there are few cultural theories that integrate conceptualizations of technology in this way.

Contributions for Theory on Technology Development

The data in this paper showed how using technology concepts as a frame led individuals to draw on their cultural resources to construct a neat problem out of a messy set of circumstances. Because ideas about technological solutions were necessary to render problems solvable, innovators at Autoworks did not experience a sequential ordering where problems first presented themselves and then the search for solutions began. Rather, in line with existing research on organizational decision making and the social construction of science, problems and solutions were dialectically implicated. Conceiving of a technology's functionality helped to guide innovators' beliefs about what was possible, and with that understanding further ideas about what a workable technological solution should look like became apparent. Technologies as frames simultaneously allowed actors to reciprocally define problems and solutions.

Research into framing strategies suggests that when changes in the environment create opportunities for action, organizations often try to mobilize others by invoking alignment strategies, which consist of bridging related but unconnected frames, amplifying latent frames, and extending a frame to a new issue (Snow et al. 1986). Benford and Snow (2000) distinguish two types of frames that are important in the early stages of new technology development. "Diagnostic frames" identify a problem that a movement should solve while "prognostic frames" involve the articulation of a proposed solution to that problem. Benford and Snow suggest that members from one group often use the tactics mentioned above to align their diagnostic and prognostic frames with the frames held by other groups. Alignment is essential so that the groups can act collectively to initiate a social movement. The framing contests that arise out of an effort to achieve alignment not only occur in the contexts of social movements, but also between groups within a single organization. Researchers who have adopted the distinction between diagnostic and prognostic framing processes have shown that when members of one group recognize that they have different conceptualizations of a problem than do members of other groups, they actively engage in contests that attempt to align their frames. This research shows that the identification of solutions (prognostic framing) is usually stalled until groups agree on the collective definition of the problem is (diagnostic framing), but after the solution is taken for granted, the groups rarely revisit whether the problem they identified was the best one (Gilbert 2006, Kaplan 2008).

In the context of the technology development process, however, the distinction between diagnostic and prognostic frames seems problematic because it implies that problems exist before solutions, and, consequently, it advocates a sequential process of active frame alignment—first groups work strategically to agree on what the problem is, and next they come to agreement on a feasible solution. The data presented herein suggest that one department's technology concept puts a frame around a specific set of cultural resources, which they then use to construct a problem. In this view, there is no distinction between diagnosis and prognosis; they are one and the same. In other words, a technological solution is in the conversational mix as early as, if not before, a proposed problem.

That innovators discuss solutions before they are entirely clear on what problems they are constructing is precisely the reason why innovation blindness occurs. Focusing on the technological artifact and its features at a granular level forestalls theoretical discussion about what aspects of the current environment are problematic in achieving a common goal. As innovators focus on solutions instead of problems, and disagree on them, they attempt to become more specific in their explanations. In this study, one innovator built a physical prototype so that he could show others how his solutions would work. Other studies have shown that innovators employ three-dimensional modeling technologies so that they can visualize the working of particular solutions (Carlile 2004), they draw blueprints and build physical parts (Bechky 2003), and they make detailed technology audits so they can "come to a similar understanding of the product" (Dougherty 1992, p. 196). However, when technological solutions are not logical outgrowths of well-defined problems but framing devices that catalyze the process of problem construction, specific grounded discussions may be more of a hindrance than a help. At Autoworks, focusing on concrete features at the granular level only compelled members of the four departments to debate more about what the technology should look like. Because they had not fully constructed problems, they shied away from openly discussing their thoughts about why they should build the technology in the first place, and let the discussion of the concrete dominate the discussion of the abstract, which ultimately left them blind to the nature of their differences.

By moving away from the concrete and reintroducing ambiguity in the process, innovators did not come to mutual understanding, but they did not let progress get mired in debate. The findings suggest that in situations where innovation blindness exists, reorganizing traditional boundaries in ways that make ambiguity a structural regularity may be one way to assure that technology development is able to continue. At Autoworks, the idea of "standardization" was itself ambiguous. However, by reorganizing boundaries in such a way as to create a new department that created standards, only one group of innovators ever had to maintain a concrete understanding of what "standardization meant," whereas the other four departments could fold their own abstract interpretations into the term without consequence. This structurally enabled ambiguity moved the development of the technology along by "packaging a compromise that 'gets the job done,' that is, that closes the system locally and temporarily so that work can go on" (Gerson and Star 1986, p. 266). The plasticity enabled by such configurations provides an ill-structured response to the exigencies of the situation that each other actor can then interpret as fulfilling his or her own needs. It is only when organizations begin to disengage from the process that ambiguous articulations can become concrete and structured in such a way as to physically build a new technology. The findings of this study suggest that reframing of the problem in an ambiguous way and restructuring the development process in a way that allows that ambiguity to be cognitively maintained are two mechanisms that organizations may use for dealing with innovation blindness.

Although the technological artifact resulting from a process that is spurred by these two mechanisms may not perfectly represent the needs of each department, or contain all of the features that would ultimately benefit users, reintroducing ambiguity at least provides a means for a technology to move forward if innovators remain blind to the nature of their differences. Because technological solutions are so intimately tied to a department's cultural values via problem construction, a clear vision of why innovators disagree on a technology's features may result in protracted cultural battles that ultimately derail the technology's development.

In closing, it does appear that the innovation process sometimes goes on in much the same way that blind men come to know an elephant. Innovators from different departments use technology concepts to frame cultural resources that lead to differential problem construction. Each innovator focuses on only one problem that must be solved to achieve a common organizational goal. To the extent that innovators cannot see that their problem is different from the problems of others, they will remain blind to the nature of their differences and, as Saxe reminds us in the closing lines of his poem, "rail on in utter ignorance of what each other mean." Hopefully, this study will be one step toward helping organizational actors realize that innovation blindness arises out of differential problem construction and is perpetuated by a myopic focus on technologies' features in cross-boundary interactions.

So oft in theologic wars,
The disputants, I ween,
Rail on in utter ignorance
Of what each other mean,
And prate about an Elephant
Not one of them has seen!

Acknowledgments

The author thanks Diane Bailey, Steve Barley, and Jeff Treem for comments on earlier versions of the manuscript. Klaus Weber and Tina Dacin also offered many helpful suggestions for improvement throughout the review process. Support for this research was given by National Science Foundation Grant ITR 0427173.

Endnotes

[1]All names used in this manuscript are pseudonyms.

[2]A solver is a code that performs computational analyses on finite element models using linear or nonlinear numerical schemes.

[3]Model instrumentation involved three essential steps: (1) Placing accelerometers on a model to indicate at what locations the solver should measure deceleration of the structure, (2) making section cuts in their model to tell the solver where to measure the forces generated during impact, and (3) positioning the barrier that the vehicle would impact.

[4]The R-drive was a drive on the supercomputing server in which analysts were allotted a certain amount of space to store

their models. The size of the R-drive allotted to analysts was determined by Infoserv and based on considerations of computing capacity and resources.

References

Alvesson, M. 2002. *Understanding Organizational Culture*. Sage, London.

Amabile, T. M. 1996. *Creativity in Context*. Westview Press, Boulder, CO.

Bechky, B. A. 2003. Sharing meaning across occupational communities: The transformation of understanding on the production floor. *Organ. Sci.* **14**(3) 312–330.

Benford, R. D., D. A. Snow. 2000. Framing processes and social movements: An overview and assessment. *Annual Rev. Sociol.* **26** 611–639.

Bowker, G., S. L. Star. 1994. Knowledge and infrastructure in international information management: Problems of classification and coding. L. Bud-Frierman, ed. *Information Acumen: The Understanding and Use of Knowledge in Modern Business*. Routledge, London, 187–216.

Callon, M. 1980. The state and technical innovation: A case study of the electric vehicle in France. *Res. Policy* **9**(4) 358–376.

Carlile, P. R. 2002. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organ. Sci.* **13**(4) 442–455.

Carlile, P. R. 2004. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organ. Sci.* **15**(5) 555–568.

Carlson, W. B. 1992. Artifacts and frames of meaning: Thomas A. Edison, his managers, and the cultural construction of motion pictures. W. E. Bijker, J. Law, eds. *Shaping Technology/Building Society: Studies in Sociotechnical Change*. MIT Press, Cambridge, MA, 175–198.

Cohen, M. D., J. March, J. P. Olsen. 1972. A garbage can model of organizational choice. *Admin. Sci. Quart.* **17**(1) 1–25.

Collins, H. M., T. J. Pinch. 1982. *Frames of Meaning: The Social Construction of Extraordinary Science*. Routledge and Kegan Paul, London.

Collins, S. T. 2003. Using ethnography to identify cultural domains within a systems engineering organization. *Bull. Sci. Tech. Soc.* **23**(4) 246–255.

Davidson, E. 2002. Technology frames and framing: A socio-cognitive investigation of requirments determination. *MIS Quart.* **26**(4) 329–358.

Davidson, E. 2006. A technological frames perspective on information technology and organizational change. *J. Appl. Behav. Sci.* **42**(1) 23–39.

DiMaggio, P. J. 1997. Culture and cognition. *Annual Rev. Sociol.* **23** 263–287.

Dougherty, D. J. 1992. Interpretive barriers to successful product innovation in large firms. *Organ. Sci.* **3**(2) 179–202.

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

Eisenberg, E. M. 1984. Ambiguity as strategy in organizational communication. *Comm. Monographs* **51**(3) 227–242.

Eisenberg, E. M., P. Riley. 2001. Organizational culture. F. M. Jablin, L. L. Putnam, eds. *The New Handbook of Organizational Communication: Advances in Theory, Research, and Methods*. Sage, Thousand Oaks, CA, 291–322.

Entman, R. 1993. Framing: Toward clarification of a fractured paradigm. *J. Comm.* **43**(4) 51–58.

Ettlie, J. E. 2007. Empirical generalization and the role of culture in new product development. *J. Product Innovation Management* **24**(2) 180–183.

Fiske, S. T., S. E. Taylor. 1991. *Social Cognition*. McGraw Hill, New York.

Gerson, E. M., S. L. Star. 1986. Analyzing due process in the workplace. *ACM Trans. Office Inform. Systems* **4**(3) 257–270.

Gilbert, C. G. 2006. Change in the presence of residual fit: Can competing frames coexist? *Organ. Sci.* **17**(1) 150–167.

Goffman, E. 1974. *Frame Analysis: An Essay on the Organization of Experience*. Harper Colophon, New York.

Kaghan, W. N., G. C. Bowker. 2001. Out of machine age?: Complexity, sociotechnical systems and actor network theory. *J. Engrg. Tech. Management* **18**(3–4) 253–269.

Kaplan, S. 2008. Framing contests: Strategy making under uncertainty. *Organ. Sci.* **19**(5) 729–752.

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

Leonardi, P. M. 2007b. Organizing technology: Toward a theory of socio-material imbrication. Unpublished doctoral dissertation, Stanford University, Stanford, CA.

Leonardi, P. M. 2009. Why do people reject new technologies and stymie organizational changes of which they are in favor? Exploring misalignments between social interactions and materiality. *Human Comm. Res.* **35**(3) 975–984.

Leonardi, P. M. 2011. When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. *MIS Quart*. **35**(1) 147–167.

Leonardi, P. M., S. R. Barley. 2008. Materiality and change: Challenges to building better theory about technology and organizing. *Inform. Organ.* **18**(3) 159–176.

MacKenzie, D. 1996. *Knowing Machines: Essays on Technological Change*. MIT Press, Cambridge, MA.

Martin, J. 2002. *Organizational Culture: Mapping the Terrain*. Sage, Thousand Oaks, CA.

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

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

Orlikowski, W. J., D. C. Gash. 1994. Technological frames: Making sense of information technology in organizations. *ACM Trans. Inform. Systems* **12**(2) 174–207.

Saxe, J. G. 1873. *The Poems of John Godfrey Saxe, Complete Edition*. James R. Osgood and Company, Boston.

Schein, E. H. 2004. *Organizational Culture and Leadership*, 3rd ed. Jossey-Bass Publishers, San Francisco.

Snow, D. A., E. B. Brochford, S. K. Worden, R. D. Benford. 1986. Frame alignment processes, micromobilization, and movement participation. *Amer. Sociol. Rev.* **51**(4) 464–481.

Star, S. L. 1989. The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving. L. Gasser, M. N. Huhns, eds. *Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, San Francisco, 37–54.

Strauss, A., J. Corbin. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd ed. Sage, Thousand Oaks, CA.

Swidler, A. 1986. Culture in action: Symbols and strategies. *Amer. Sociol. Rev.* **51**(2) 273–286.

Swidler, A. 2001. *Talk of Love: How Culture Matters*. University of Chicago Press, Chicago.

Trice, H. M. 1993. *Occupational Subcultures in the Workplace*. ILR Press, Ithaca, NY.

Van de Ven, A. H. 1986. Central problems in the management of innovation. *Management Sci.* **32**(5) 590–607.

Van Maanen, J., S. R. Barley. 1984. Occupational communities: Culture and control in organizations. L. L. Cummings, B. M. Staw, eds. *Research in Organizational Behavior*. JAI Press, Greenwich, CT, 287–365.

Yin, R. K. 1994. *Case Study Research: Design and Methods*. Sage, Beverly Hills, CA.

Paul M. Leonardi is an assistant professor in the Departments of Communication Studies and Industrial Engineering & Management Science at Northwestern University, where he holds the Allen K. and Johnnie Cordell Breed Junior Chair in Design. He received his Ph.D. from Stanford University. His current research explores how organizations can employ advanced information technologies to more effectively create and share knowledge.

Copyright 2011, by INFORMS, all rights reserved. Copyright of Organization Science: A Journal of the Institute of Management Sciences is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.