Coordination in Innovative Design and Engineering: Observations from a Lunar Robotics Project

Laura A. Dabbish[a], Patrick Wagstrom[b], Anita Sarma[c], and James D. Herbsleb[a]

[a]School of Computer Science
Carnegie Mellon University
5000 Forbes Ave,Pittsburgh, PA15213
{dabbish, jdh}@cs.cmu.edu

[b]IBM TJ Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532
pwagstro@us.ibm.com

[c]Department of Computer Sci. & Eng.
University of Nebraska, Lincoln
Lincoln, NE 68588-0115
asarma@cse.unl.edu

ABSTRACT

Coordinating activities across groups in systems engineering or product development projects is critical to project success, but substantially more difficult when the work is innovative and dynamic. It is not clear how technology should best support cross-group collaboration on these types of projects. Recent work on coordination in dynamic settings has identified cross-boundary knowledge exchange as a critical mechanism for aligning activities. In order to inform the design of collaboration technology for creative work settings, we examined the nature of cross-group knowledge exchange in an innovative engineering research project developing a lunar rover robot as part of the Google Lunar X-Prize competition. Our study extends the understanding of communication and coordination in creative design work, and contributes to theory on coordination. We introduce four types of cross-team knowledge exchange mechanisms we observed on this project and discuss challenges associated with each. We consider implications for the design of collaboration technology to support cross-team knowledge exchange in dynamic, creative work environments.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces– *Computer supported collaborative work.*

General Terms

Design, Human Factors. Management

Keywords

Engineering, design, teams, groups, coordination, collaboration, communication, intergroup collaboration, multi-group project.

1. INTRODUCTION

Engineering projects in both industry and research settings are typically comprised of several groups working collaboratively to design and build a new system [6, 7]. Groups on these projects represent different functional areas and disciplines that work together to generate innovations and combine technology in new ways. In this context defined by concurrent work, rapid iteration, and functional diversity, teams are faced with the challenge of coordinating and integrating their efforts. Coordination among the many parties involved is critical to project success and difficult to achieve [1, 2, 5, 14, 21, 27]. However, our understanding of boundary-crossing coordination in dynamic design teams is in fact quite limited [1, 18].

In order to design systems that support cross-team collaboration in highly dynamic, creative settings such as engineering research and product development, we need a better understanding of how work is coordinated across groups. Thus, our goal in this research is to understand the nature of intergroup coordination on a dynamic and emergent engineering research project. Towards this goal, we conducted an in-depth field observation of collaboration in a project developing a semi-autonomous lunar exploration rover robot. The project was comprised of several teams, each organized around a distinct functional area of the work.

We chose to observe coordination on this project because the work of each subgroup was innovative, emergent, non-routine, and highly dependent on the work of other subgroups. The mechanical team on the project worked on a digital design of the next generation version of the lunar exploration rover at the same time that hardware and testing teams fabricated and tested the current generation prototype. Figure 1 shows an example of a current generation physical prototype created and tested by the team during our observation (left) while the next generation prototype (right) was being designed. Results on the performance of the current generation prototype would continuously feed back from the testing team to influence the mechanical team's design of the next generation version of the robot. The hardware team would make adjustments to the current generation physical prototype to test out the mechanical team's design ideas for the next generation prototype.

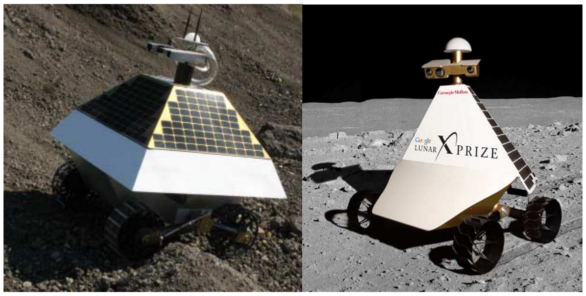

Figure 1. Left: Physical prototype of second-generation rover being tested in the field. Right: CAD diagram of third-generation rover.

GROUP'10, November 7–10, 2010, Sanibel Island, FL, USA.

In addition, there was a high level of uncertainty involved in what the final product would look like. The rover was a novel system being designed to function semi-autonomously in an environment where the teams did not have extensive experience. The project members were required to constantly innovate to solve problems unique to the constraints of the project, as they learned about the nature of space flight and the demands of the lunar environment. Their work was characterized by high levels of interdependence, dynamism, and uncertainty. It is difficult to coordinate activities in this type of setting because you cannot anticipate in advance what the details of the work will look like, map out dependencies a priori, or know exactly what information or resources will be needed to complete the work [14, 15, 28, 42].

In the context of this dynamic, highly interdependent, innovative engineering research project we considered the following questions: When and how do project members exchange knowledge across group boundaries? What is the nature and content of the interactions that cross group boundaries? What issues and problems arise in coordinating activities across groups? Our goal was to describe knowledge transfer mechanisms that facilitate cross-group coordination and inform the design of collaboration technology for innovative design work.

We observed that collaboration across functional groups on the project consisted of selecting and translating the knowledge products generated within each group. Based on our observations, we identified four types of cross-group knowledge translation mechanisms. We discuss consequences we observed that were associated with cross-group knowledge translation and consider potential implications for the design of collaboration technology to support cross-team interaction on a project.

2. RELATED WORK

2.1 Coordination Theories

Coordination involves managing dependencies among tasks [31]. Tasks may be interdependent because they share resources, have an input output relationship, simultaneity constraints, or task subtask relationships [31]. Our current understanding of coordination has its origins in organizational design research, which conceptualizes the organization as a response to task and environmental uncertainty. In this view, appropriateness of coordination mechanisms (formal vs. informal) depends on the level of uncertainty [15, 32, 41]. Formal coordination mechanisms, appropriate when uncertainty is low (e.g., routine tasks), involve a priori definition of organizational structures and processes for managing dependencies including supervision, rules, routines, standardization, scheduling, pre-planning, and division of labor into minimally dependent units [32, 41].

Informal coordination mechanisms, on the other hand, involve ad hoc coordination by mutual adjustment- meaning individuals directly interact to exchange task status information and negotiate dependencies. These interpersonal coordination mechanisms are better suited for managing highly interdependent, complex tasks because of their higher information processing capabilities [12, 15, 31, 32, 41]. At the same time, in organizations characterized by rapid change, these types of informal mechanisms are prohibitively costly [41]. It is not reasonable or currently feasible, for example, for every engineer involved in a project to talk to every other engineer each time a change is made. Thus the organizational design theories of coordination provide only very high level guidance, in part because task dependencies are expressed in terms of stable, coarse-grained, stylized patterns that must be known in advance.

More recent work on coordination in highly interdependent complex environments has begun to examine the nature of informal coordination in more detail [14, 16, 25, 27, 30, 33]. In a recent review, Okhuysen and Bechky [36] aggregate much of this more recent work noting that key informal mechanisms for coordinating in emergent environments support mutual awareness of who is responsible for specific elements of a task, predictability about subsequent task related activity, and a shared perspective on the task as a whole. In dynamic settings where new knowledge creation is the focus, such as in new product development environments [3, 7, 8], and space mission planning [33], knowledge exchange interactions are critical for supporting integration of the knowledge being generated [9]. These take the form of either communication or the exchange of artifacts supporting information sharing, activity organization, awareness of ongoing work status, mutual adjustment around conflicts, and creating a shared understanding [4, 11, 29, 36]. Because of the recognized importance of knowledge exchange interactions for coordination in emergent, dynamic environments, in the current study we focused specifically on capturing and describing knowledge exchange interactions that occurred across groups to support dependency management.

2.2 Engineering and Product Development

Research in engineering management and product development domains has highlighted integration of the work of subgroups as a critical project success factor [6, 7, 21, 28, 42]. Engineering and new product development projects require collaboration among individuals and teams from a variety of functional disciplines including marketing, design, and manufacturing [6, 7]. Studies in these settings have found that, in general, more cross-team communication and boundary crossing interaction improves performance [1, 24, 25]. However, these studies have considered product development teams in place over a long term with more extensive use of formal coordination mechanisms. When teams are creating an innovative product, there are high levels of uncertainty involved [7, 34]. Coordination research in these settings has suggested that frequent direct communication is essential for facilitating updates on product status [40], but has only begun to describe in detail the nature of knowledge exchange interactions that occur across disciplinary units or teams in such an adhoc environment [7, 8]. Work by Bergman [4], Carlile [7], Christensen [11], and Lee [29] has highlighted the importance of objects and representations, particularly in the form of project plans, diagrams and drawings for facilitating the transfer of knowledge in design work. Carlile and Rebentisch [9] suggest that the process of transforming knowledge is critical for coordinating the activities of specialized sub-groups. However, this previous work has not considered the transfer of knowledge as part of an ongoing iterative development project where multiple interdependent groups are simultaneously generating knowledge. Knowledge transfer is often viewed as the endpoint of a particular group's work and the starting point of the next groups work. In our observations we sought to extend the previous work on coordination in product development by considering what knowledge exchange interactions look like when subgroups in a project are working in parallel, and knowledge exchange is needed to support ongoing activity within each group.

2.3 Software Integration

Cross-team coordination has also been a topic of much consideration in the software development domain [13, 18, 22, 23]. Large software projects are often divided into modules that are developed by several different teams. The common wisdom is that coordination across teams should be managed using programmed interfaces between the modules of the code being developed [18, 23]. Despite extensive upfront planning, however, integration can still pose difficulties as requirements and processes evolve over the life of the project. Herbsleb and Grinter [23] found that a priori integration plans are often not followed and interfaces between components have to be renegotiated because of changes to requirements between the planning and integration phases. In their observations, Herbsleb and Grinter [23] also found that integration was complicated by unique development processes that evolved within each team. In software, as compared with engineering, the nature of work within each team is somewhat comparable and the toolset in use is often consistent across projects. We expect subgroup differences in practices to vary more widely and pose even more challenges in an engineering project organized around functional areas.

2.4 Knowledge Sharing

Research on organizational communication and knowledge sharing has considered the nature of knowledge exchange interactions in information-centric work settings. This research has found that communication across boundaries facilitates the transfer of knowledge, information, and expertise beyond the department or organization [5, 24, 35] and that there are unique considerations associated with communication across disciplinary boundaries. The know-how developed within a functional unit or a community of practice can be very "sticky" or resistant to transfer because it is context-specific, borne out of repeated interactions among community members, and based on a shared understanding and shared sets of values within a domain [5, 24, 35]. This kind of embedded expertise can be difficult to share with others from a different organization or disciplinary perspective. In an engineering project, subgroups also face the same type of challenge in transferring knowledge to other subgroups with different disciplinary perspectives [6, 7, 25, 26]. Thus, they may be likely to experience difficulty communicating and collaborating across group boundaries, despite having a shared goal at the project level [7]. "Boundary objects", shared or shareable objects or artifacts that establish a shared context across disciplinary or functional boundaries, can facilitate collaboration by supporting the transfer of knowledge between groups [7, 39]. Given the importance of "boundary objects" for supporting cross-boundary knowledge exchange, in our observations we noted in particular communication and coordination that occurred around electronic and physical artifacts shared between subgroups.

In is important to note that much of the previous work on knowledge transfer has focused on interactions across departmental or organizational boundaries, where participants may not necessarily have a common interest or shared goal. When multiple teams are collaborating on the same engineering or product development project they are embedded within the social unit of the project. They have a shared goal at the project level which can function to coordinate actions across disciplinary boundaries as Star and Griesemer [39] observed. Teams may be able to align their cross-team activities around the shared purpose of the team as a whole.

We sought to understand and describe collaboration across teams in an engineering project by studying in detail the work process on the lunar robotics project. We next describe our data collection procedures on the project observed, and then review our key findings about knowledge translation across team boundaries, and associated consequences of different translation mechanisms.

3. CASE STUDY: LUNAR ROVER DESIGN

3.1 Site and Project

We conducted a 12 month in-depth field observation of a privately funded engineering organization competing in the Google Lunar X-Prize challenge [17] to send a mobile robot to the moon. Competing teams need to launch a robot into space and land it safely on the moon. The robot needs to traverse 500 meters on the moon's surface, capturing and transmitting high definition video back to earth. The first team to meet the goals by December 31, 2012 will be awarded $20 million dollars.

We observed interactions within an organization responsible for the lunar rover design portion of this project. The goal of the project was to design and fabricate a robot that could complete the mission requirements and withstand space flight, moon landing, and the moon's harsh environment. The organization observed was a world leader in field robotics but had minimal experience with interplanetary systems, space flight, and the lunar environment. The project involved a combination of a handful of very experienced robotics engineers (+20 years) and many recent college graduates with a variety of engineering backgrounds.

In the lunar rover design project, work was occurring primarily at one site of one organization, involving a (mostly) collocated set of 24 engineers (4 remote) from various disciplines including mechanical, optics, hardware, and software. The four teams at this site were organized around each discipline, with a technical lead for each team. Each team on the rover project was responsible for a different aspect of the rover's creation. The mechanical team was responsible for primary structure design, and the optics team was responsible for camera system selection and design. The hardware team was responsible for constructing and testing physical models of the rover, also referred to as rover prototypes. The software team was responsible for writing the software to drive rover functionality. The project used iterative design, build, and testing stages, as described in the introduction, where the next prototype of the rover was being designed while the current prototype was being tested.

There were also several engineers who were not embedded in a team: the thermal engineer, electromechanical engineer, communications engineer, and overall program manager. The first three of these engineers were responsible for thermal analysis, electrical component selection, and communications system design respectively. The program manager was responsible for creating systems requirements and writing a mission plan for what the rover would do while on the moon. Engineers at the other sites were primarily involved as key technical resources, advising the team and reviewing their progress.

Our data collection on this project was designed to capture in detail the nature of work within each team, and the collaboration and communication practices across the teams. We conducted meeting capture, observation, project member interviews, and artifact analysis.

3.2 Meeting Capture

During our observation period, we video recorded in-person meetings and audio recorded technical conference calls, taking detailed notes while meetings were taking place. This included weekly project-wide status meetings at the lunar rover design site, bi-weekly conference calls with the other sites, and occasional longer technical review sessions focusing on specific issues.

3.3 Observation

We periodically visited the different working locations associated with the project at other times when no meetings were taking place. During these times we observed team members working, sharing resources, and were able to informally discuss our interpretations of project activities and technical details with different project members.

3.4 Member Interviews

In addition to observations, we conducted semi-structured interviews with every project member. In the interviews, project members were asked about their role on the project, the work they had done in the preceding week, and how they interacted with other engineers to accomplish that work. We took detailed notes and recorded audio. These interviews supplemented our direct observations of project activity and gave us a more complete picture of activity within and across the subgroups of the project.

3.5 Artifact Analysis

The project utilized a number of computer-mediated communication tools to share files, distribute meeting reminders, and disseminate other project-relevant information. We augmented our observations and interviews with messages from project mailing lists, website postings, design documents, test plans and results, and engineering specifications.

We aggregated our data from meetings, observations, and member interviews to describe in detail the work done within each team and the information, communication, and artifacts shared across teams. We identified instances of cross-team collaboration, and for each instance we detailed what information was shared and how. We analyzed these examples of cross-team collaboration by looking for cases that shared common characteristics and cases that differed. This analysis helped us identify general methods of coordinating work between teams and their defining attributes. Based on this qualitative analysis of the data gathered across our 12 month observation period, we were able to clearly distinguish four unique mechanisms for selecting and translating knowledge products generated within each team across team boundaries. In the next sections, we describe the four cross-team knowledge sharing mechanisms we identified, and then discuss the consequences we observed associated with the cross-team knowledge sharing process.

4. CROSS-TEAM COORDINATION AS DYNAMIC KNOWLEDGE SHARING

For each of the teams on the project, knowledge inputs from other teams were primary drivers of their work activities. For example, the components in the mechanical team's design were based on functional requirements generated by the program manager. And they iterated and refined their design based on the testing team's results and feedback from field tests of the robot prototype or bench tests of individual components. The thermal engineer conducted modeling based on the component system designs generated by the mechanical team. In addition, her models of thermal limits in the moon's environment drove the testing team's development of equipment for thermal testing purposes. These types of dependencies were present across all of the teams on the project.

We found that cross-team collaboration on the project generally took the form of *selecting* knowledge thought to be useful to another team, and *translating* it into a form that the other team could use in their work. By analyzing the set of knowledge sharing instances drawn from our observational data, we identified four types of mechanisms that were used for cross-team knowledge sharing:

- *Pushing communication artifacts*
- *Extracting knowledge*
- *Interactive sharing*
- *Direct collaboration*

These mechanisms varied in terms of what initiated the knowledge sharing, who did the work of selecting and translating the knowledge, how sharing occurred, the type of knowledge products shared, and issues that arose in the sharing process.

4.1 Pushing Communication Artifacts

One way that knowledge was transferred across teams was through the creation and exchange of communication artifacts. These artifacts represented a selection of knowledge a team thought would be relevant to another team, and had translated into a usable and codified form. These communication artifacts, once generated, functioned to some extent as "boundary objects" [4, 7, 29, 39] identified in previous work as objects that establish a shared context between domains. The cost of creating these artifacts fell primarily on the team in possession of the knowledge (knowledge generators). Exchange was typically accomplished via electronic means, by posting the communication artifact to the team website, emailing the artifact, or posting it on a shared server. The original generator of the knowledge would select and translate the knowledge into a communication artifact and the team consuming the knowledge would often need to translate the artifact again to use it internally. This was typically done when one team had completed some aspect of their internal work and wanted to communicate accomplishments or milestones, when one team needed to communicate instructions or plans to another team (i.e. when one team needed another team to do something on their behalf), or for tracking information across multiple teams. We witnessed the use of several such artifacts for transferring knowledge across teams. We consider examples of each of these initiators of 'pushing communication artifacts'.

4.1.1 Transmitting commands

One example of transmitting commands via a communication artifact was development of a project assignment spreadsheet. The program manager developed this spreadsheet by translating functional operations in the mission plan (e.g. approaching the Apollo 11 landing site for a photo op) into a listing of physical components that would go on the next generation rover (e.g. the mobility system). The existing prototype design was reviewed to determine what mission operations it could not yet perform.

Table 1. Knowledge Exchange Mechanisms Observed

Exchange mechanism	Description	Form of sharing	Who was involved in selection and translation	When was this done	Issues that arose
Pushing communication artifacts	Creation of an artifact (image, document, spreadsheet) that codified the state of knowledge within a team	Artifact exchange, often conducted electronically	Knowledge generator would select and translate knowledge	When knowledge generator needed knowledge consumer to do something on their behalf	Effort required in creating artifacts explicitly for communication purposes
Knowledge extraction	Member of another team selected knowledge artifacts from another team and translated for their own use	Artifact extraction, followed by artifact modification	Knowledge consumer would select and translate knowledge	When one member needed latest state of knowledge output for their own work	Difficulty understanding and translating artifacts that were not created for consumer's purpose
Interactive sharing	Interactive communication focused on exchanging current state of knowledge across teams. Generator would present state of knowledge, and consumer would ask questions.	Formal meetings (e.g. design reviews) teleconferences, or informal face to face conversations around electronic or physical artifacts	Knowledge generator would select prior to interaction, generator and consumer would interactively translate	When one team had reached a stable point in the state of their knowledge.	Not archived, and not all generators and consumers could be present. Issues that arose during knowledge consumption were not always anticipated during interaction.
Direct collaboration	Members from different teams would work together on a joint task.	Face-to-face communication and tacit coordination during task execution.	Knowledge generator and consumer interactively select and translate knowledge to complete the consumer's task.	When generator wanted to know consumer task output or was available in close proximity	Most available member from generator team may not be original knowledge generator

The mechanical lead then used this spreadsheet to assign the design of components to members of the mechanical team. Because the mission plan was fairly stable over time, these spreadsheets were created at the beginning of the design phase for the next generation prototype of the rover (i.e. every 3-4 months).

Another observed instance of a communication artifact for transmitting commands was creation of the software specifications. The software specifications were created by the mechanical lead and electromechanical engineer. These engineers translated the mechanical team's project assignment spreadsheet into a specification document that the software team could work from. Thus the software development work was also based on the knowledge in the mission plan, as "re-translated" by the mechanical lead. This translation was done once.

The field test checklist was another communication artifact for transmitting commands, also generated based on the mission plan. This text document was used to run the field test operations. The document listed operations the prototype robot would attempt during each field test, where the robot would be taken to an offsite testing facility and run through a mock "mission". The checklist was created for use by the software and hardware teams based on the program manager's mission plan. Creating the checklist translated knowledge in the mission plan into a format useful for the field test activities.

4.1.2 Communicating status

The mechanical team's CAD diagrams represented another instance of communication artifact creation. The CAD models represented the current state of the rover design as the mechanical team saw it, but changed rapidly due to the highly iterative nature of work within the team. Mechanical team members generated a selection of jpg images from the models to support cross-team dissemination of the most recent status of their design. When the team finished designing a particular component or drastically modified the design of a particular component, images of the component design would be posted to the team website or displayed at the team's weekly status meetings to disseminate awareness of the latest design changes.

4.1.3 Tracking decisions across groups

Communication artifacts were also used for centralizing information about the state of knowledge in different subgroups. The team used a spreadsheet they called a "vendor book" to track equipment purchasing at the project level. For each piece of equipment needed, team members entered different options under consideration, vendors providing each option, and pricing. The vendor book then acted as a communication artifact with project wide equipment and purchasing information. The book represented a selection and translation effort by each team, as they converted their tacit information about the vendors and equipment they were considering into an explicit, codified representation of relevant information such as component, vendor, and price. Because of the effort needed, the project manager and overall project leader had to issue repeated mandates for people to contribute to this project-wide information repository.

4.2 Extracting Knowledge

In some cases, a member of one team would both select and translate knowledge from another team for their own uses. The cost of translation in these cases fell squarely upon the knowledge consumer, who had to select the right knowledge artifacts from another team and do the work to modify the knowledge into a usable form. This seemed to happen in the course of task execution, as a byproduct of the flexibility and openness of the project combined with the high levels of time pressure. When the knowledge consumer came to a point in their task where they needed knowledge generated by another team in order to complete their own work, if they knew where the knowledge was located, it was acceptable and encouraged to access and use it.

For example, we observed that the thermal engineer often extracted knowledge created by the mechanical team in order to do her own work. The design software used by the mechanical engineers had a different file format than the thermal analysis software. This meant the thermal engineer had to manually translate the CAD diagrams into the input format needed by her thermal modeling software. She was able to perform this translation of their diagrams in a fairly independent way without querying for information because she was seated in the same room with the mechanical engineers and so had developed tacit knowledge about the design and where it was located from overhearing their work.

4.3 Interactive Sharing

Knowledge was also selected and translated across teams interactively. Engineers from different teams would talk face-to-face around the knowledge being exchanged to establish a shared understanding and ensure the appropriate information was transferred. In interactive sharing the knowledge generator selected and presented a subset of information believed to be useful for the work of the other team, often in the form of an electronic artifact. Through conversation and interaction the knowledge generator 'translated' the information for the knowledge consumer, establishing common ground about what the artifact represented. Knowledge consumers could improve on the translation by expressing confusion or asking for clarification. We observed that this type of interactive sharing often occurred at stages when one team had completed their part of the work process and was ready to hand off what they had done to another team. When the knowledge being handed off was larger in scale, the interaction was accomplished through formal meetings, and when smaller in scale through periodic informal communications.

4.3.1 Formal meetings: Design reviews

Design reviews were one example of formal meetings where interactive sharing occurred. The project held design reviews when the mechanical and electrical designs had stabilized in order to prepare for creation of the next generation physical prototype. Each physical prototype built represented a snapshot of the design at one moment in time (e.g. the 2nd vs. 3rd generation designs shown in Figure 1). Before a new prototype was built, the team held several extensive design review meetings lasting 8-10 hours each, where the mechanical team and electromechanical engineer described every piece of the design. The hardware team was responsible for fabricating the rover prototypes using the design plans being presented. Because of this, the hardware team was extremely interactive during the design reviews, asking questions to clarify any ambiguities in the diagrams and images presented. They also considered carefully the problems that might come up in their own work when viewing the design. As each piece of the design was presented they discussed fabrication feasibility and did not hesitate to point out potential problems they saw. After the meeting the hardware team would work to fabricate the prototype based on the new design, with periodic questions to and feedback from the mechanical and electromechanical engineers.

4.3.2 Informal interactions: Test plans and results

Informal or unplanned face-to-face interactions also supported interactively sharing knowledge products of one team with another team. For example, the mechanical team's next generation rover design was informed by test results that were interactively shared at regular intervals by the hardware team. Members of the mechanical team often requested that certain tests be run with particular equipment configurations to help them flesh out their design. Excel spreadsheets were used by the testing team to support record keeping for the tests that were conducted, and test reports were created to share the results. After the tests had been run, the results were communicated to project engineers face-to-face, with the reports, videos, and photos acting as supporting documentation. Figure 2 shows video footage from component testing of the robot's sidearm and wheel configuration. Testing team members would show mechanical team members pieces of footage from these videos to demonstrate problems that occurred during testing, and supplement the video with comments and gestures indicating what they had observed as mechanical issues. The mechanical team members would ask follow up questions to get more detail about the test results, for example asking about the testing environment or the behavior of specific mechanical components.

Figure 2. Video taken during a subsystem test.

4.4 Direct Collaboration

Knowledge was also shared across teams through direct collaboration between members from each team. Rather than scheduling or finding time for interactive sharing sessions, engineers from different teams would work together on their respective engineering tasks. While directly collaborating, team members could rapidly select the most relevant information to exchange in the moment and translate it into task-relevant conversation or action through their synchronous face-to-face interactions with the other team member around the shared task.

4.4.1 Physical collocation and flexibility

Direct collaboration was facilitated by flexible physical arrangements allowing the engineers to collocate when necessary. This was further reinforced by a norm of openness around information and task responsibilities. The project members valued getting things done above all else, and exhibited almost no territoriality or possessiveness about responsibilities, resources, or information. Several engineers even acted as unofficial members of teams outside of their main one, frequently interacting with individuals on those teams when their work became highly interdependent or a problem arose that affected both teams.

One example of such boundary spanning was the optics lead GK, who would collocate with other teams when his work became highly interdependent with theirs. GK and one of the software engineers, FM, directly collaborated to create and test the software for the camera, translating knowledge in both directions. The camera software was supposed to capture and process images and video taken by the rover camera. Creating this software required a detailed understanding of the camera design in order to ensure the compression and transmission algorithms could satisfy the mission requirements with the available bandwidth. GK and FM worked together to determine whether the camera system and software was functioning correctly by performing unit tests of the imagery software independent of the weekly field test activities. GK also attended the software teams weekly "testing and development" meetings and became an almost defacto member of the software team, often sitting in their work area in the "highbay", a large open equipment testing room.

GK also directly collaborated with the mechanical lead (TJ) to design housing for the camera. The optics team needed to understand the latest status of the CAD design from the mechanical team to create their own design of how the camera would be housed on the rover. GK was often collocated with the mechanical team in the "project room," so whenever GK needed to interface with TJ, TJ would simply wheel over to GK's workbench, where they would discuss and edit the design. Once both engineers were satisfied, TJ would return to his workstation.

4.4.2 Shared interest in task results

Direct collaboration was often spurred by shared interest in the result of a task because of a dependency between two tasks assigned to different teams. For example, the mechanical team and hardware team often directly collaborated to run unit tests evaluating design options for a particular component. This was motivated both by the mechanical engineer's desire to see a prototype of their design in use, and the hardware team's desire to understand how the component should be built. Mechanical engineers would frequently work directly with the hardware team to construct the prototype component for a test and create the right test setup. Through this collaboration, mechanical engineers would interactively share knowledge about the component design and hardware engineers would share knowledge about the assembly process. For example, the mobility engineer, HS, was considering two wheel design options, a solid versus a treaded wheel, and wanted to test their respective abilities to drive over obstacles. He set up a test with the hardware team and worked directly with them to build a mockup of his planned wheel designs at "robot city" and test obstacle clearance abilities against obstacles of different sizes. Because he was present at the test itself he directly observed the issues with his current wheel design and was able to make modifications to the CAD diagram based on what happened during the test. These kinds of task-focused collaborations between the mechanical team and the hardware team around unit testing served to seamlessly transfer knowledge about the design and the tests themselves.

5. BREAKDOWNS IN CROSS-TEAM COLLABORATION

We observed several kinds of breakdowns in the cross-team knowledge sharing process, due primarily to three factors: (1) sharing costs, (2) errors introduced in the process of sharing, and (3) concurrency conflicts. We discuss each of these issues below, and consider implications for collaboration technology design that might lower the costs associated with translating knowledge across team boundaries.

5.1 Sharing Costs

Selecting and translating knowledge to a form usable by other teams took extra time and effort on top of the work going on within teams. In our observations this effort fell on the knowledge generator, the knowledge consumer, or both depending on the mechanism used. Generating communication artifacts, for example, put a burden on the knowledge generator because they had to work to externalize knowledge in their head, or translate a selection of it into a form someone else needed. This work is something team members avoided if possible. As a result, communication artifacts were often out of date or didn't represent the true state of a team's knowledge. This created problems down the line because the work was tightly coupled across teams. Previous work on knowledge management has identified this cost of 'codification' as a major barrier for most knowledge management initiatives within corporations. However, this barrier has unique implications in ongoing design projects, where out of date information can result in coordination problems.

5.1.1 Aggregating information across teams

One example of a breakdown caused by the cost of sharing was the lack of updates to the project's master equipment list or MEL. The MEL was a large Excel spreadsheet that contained a listing of all mechanical and electromechanical subsystems on the rover, and different design options being considered for each subsystem. The MEL also recorded features of each component that needed to be tracked at the project level, such as the mass, power consumption, and price. The electromechanical engineer, MR, was the "keeper" of the MEL, but updates and changes were done on a very ad hoc basis. Project engineers would often query MR for the contents of the MEL but not provide updates or changes. For example, GK, the optics lead said he did not interface with the MEL. He largely had all the camera specifications and different viable options "in his head" and once he had finalized a component, he would let the appropriate person know so that they could update the MEL. MR would revise the MEL prior to periodic design reviews the team held every 3-4 months.

This lack of updating led to problems. In particular, several global constraints had to be met in the design of the rover. Because the MEL was not kept up to date, it was not possible to track how the project was doing with respect to these global constraints. This led to radical emergency redesigns when the MEL was updated and project members realized that constraints were violated. In one instance the MEL was updated and the team realized that the system mass was 18GK over the limit. This problem had evolved slowly over time, because of the lack of updating of the MEL, meaning tracking the problem was nearly impossible. As a result the project contemplated radical redesigns to solve the mass problem, such as removing entire subsystems of the rover.

5.1.2 Maintaining awareness of state changes

One of the challenges associated with sharing knowledge across teams was difficulty maintaining awareness of the latest state of knowledge in a different team. This was frequently an issue with the mechanical design. Because of the costs associated with sharing the latest version of the rover's mechanical design, it was not frequently done. There was no procedure in place for synching the changes to different aspects of the design or frequently communicating the status of the design:

> HS: *"There's not a very straightforward process right now [for fixing the design]. It's just been everyone in the mechanical team just working through different issues and iterating the design over and over."*

Remote members from the other project sites had the most difficulty staying updated on the design or accessing the latest version when needed. For example, RF was an engineer from a different organization responsible for guidance and navigation software for landing the robot on the moon. He was aware that the CAD server contained an obsolete version of the design and ended up polling all of the mechanical team members via email to track down the latest version. Because of the state of the CAD server, RF and other remote team members working on the other projects did not bother to learn how to access the server. This eventually led to problems when the rover team wanted to use the server as a repository and shared resource for the design reviews involving external organizations.

5.2 Sharing Errors

Another important issue we identified was that errors were sometimes introduced in the sharing process. Successful sharing required selection of the right knowledge to transfer, an accurate understanding of the knowledge originally generated and appropriate translation of the knowledge into a new format useful to the knowledge consumer. Problems could occur at every stage of this process.

5.2.1 Extracting knowledge from working artifacts

Challenges often arose in utilizing artifacts created by other teams for purposes different than that of the knowledge consumer. In particular, to perform knowledge extraction, the consumer needed sufficient understanding of the knowledge embedded in the artifact to translate it correctly. Without this understanding, consumers had to apply their best guess about the assumptions used during knowledge generation. When these assumptions were incorrect, translation errors could occur. One of these types of errors occurred when the thermal engineer manually translated the mechanical team's design. In one of the project-wide weekly meetings, she reported discovering that her translations of the CAD model specifications into her thermal modeling program had been off by a small amount, and as a result her analyses on the thermal affordances for the primary structure and wheel assembly had to be completely redone.

5.3 Concurrency Conflicts

Translating knowledge to a form usable by other teams also took time. This became a challenge because knowledge generation within each team happened concurrently and knowledge products quickly became obsolete. As each team conducted their work, new insights were generated and within each team, changes were made to design plans, code, physical models, and mission plans. But there was a sense that the internal workings of the team weren't useful until they were "finished" or "complete." This meant that different teams were often concurrently acting on and making incompatible changes to interdependent aspects of the design without knowing it.

5.3.1 Current prototype vs. next generation design

One example of concurrency conflict resulted from the parallel engineering process in use on the project. The mechanical team was engaged in designing the next generation prototype of the rover while the hardware and software teams tested the existing prototype. This style of concurrent engineering was intended to parallelize design and test activities so that the mechanical design team could improve their design based on results that were arising from the field tests. However, the mechanical design iterated so rapidly that the physical prototype of the rover quickly diverged from the next generation design, limiting its testing utility.

For example, during our observation period the current physical prototype of the robot was designed so that solar cells would surround the entire robot body. This design meant that the robot did not need to turn in place to charge as the sun moved across the lunar sky. Soon after the prototype was built, the mechanical team and thermal engineer realized that with this design the robot would get too hot because it could not release enough of the heat it would capture from the sun and the ground if solar panels were on all sides. Thus, the design was modified so that solar panels covered only half of the rover body, and the other half was left open with a large panel on the opposite side to radiate heat. This change in the design was quite significant and resulted in a complete reconfiguration of the rover chassis body. It also meant that the existing physical prototype of the design was quite different from what was now being planned, in terms of layout, center of gravity, etc. These discrepancies between the existing physical prototype and the next generation rover design limited the extent to which the mechanical team could learn from the field tests of the prototype.

5.3.2 Interdependent components

Timing was also an issue in developing the mechanical design of interdependent components of the rover. If team members' work was highly interdependent they would use direct collaboration when possible to complete tasks. However, this required explicit awareness of potential dependencies and availability of members from each team.

It was not always possible to directly collaborate on highly interdependent components, and in these cases problems sometimes arose when conflicting designs were created in parallel. For example, the optics team was designing a glass shield that would encase the camera equipment to protect it from the heat of the sun. At the same time the mechanical team was developing the mast and pan and tilt mechanism where the camera would mount. For the mechanical engineers to communicate to everyone else the full implications of the way they had designed the camera to move around would take time and the information would quickly become stale during a period where the design was changing by the hour. At the same time, the two designs were tightly interdependent. When it came time to integrate the designs the teams realized that the way the pan and tilt mechanism had been created and designed provided insufficient spacing between the rover chassis and the camera heat shield. The heat shield would collide with the rover chassis if the camera tried to tilt up or down. This forced a redesign of both components.

6. CONCLUSION

Our work has brought to light a number of key features of cross-team collaboration critical for the design of technology to support diverse interacting groups. In particular, translating knowledge across team boundaries and ensuring accurate use of the knowledge requires time and energy. This cost can become problematic when the work is dynamic and the state of knowledge is rapidly changing. It is important to consider carefully when and how the knowledge generator and consumer are involved in the translation process. If they are not able to interactively make sense of knowledge products, problems may results down the line as knowledge is used, and interaction may eventually become a necessity. Finally, on a dynamic project, timeliness is a critical consideration. We consider each of these issues below.

6.1 Knowledge Sharing is Costly

Sharing knowledge extracts a cost, since someone must do the work of selecting what knowledge is relevant for a particular need, and translating that knowledge into a form that the receiver can use. We observed that the difficulty of selection and translation tasks vary dramatically, depending on the degree of knowledge domain overlap between the generator and consumer and the degree to which the knowledge is implicit, or embedded in its context. In the simplest case, the cost can be as low as taking a few moments to answer a question or finding and sending an existing document. At the other extreme, it may take a lengthy collaboration where knowledge is gradually transferred over days, weeks, or months.

Good tool design can reduce the cost of sharing by automatically extracting and transmitting relevant changes in the state of knowledge, and by supporting asynchronous interaction around knowledge artifacts. For example, the Project path webtool, used as a repository by the team we observed, also had a discussion capability that was used heavily by the team. So one could upload, say, a design diagram, and engage others (or be engaged by others) in an online discussion directly linked to the artifact. Similar functionality is ubiquitous in social software such as blogs and wikis. Artifact repositories provide ready access to a range of communication media – text, voice, video – which could link directly to the design to facilitate low cost asynchronous sharing.

6.2 It Matters who Pays the Cost

As pointed out by Orlikowski [37] and Grudin [19], the incentives for collaborating and sharing knowledge have a strong influence on if and how collaboration occurs. When the burden falls on the knowledge generator, knowledge sharing is less likely to happen voluntarily or consistently. When the burden falls on the consumer, who needs the knowledge, the motivation is intrinsic. As we observed, when everyone is both a generator and consumer, as of the MEL and other artifacts shared across the project, this can lead to a sort of tragedy of the commons, where everyone would be better off if the MEL were always up to date because everyone relies on the information it contains, but it is not worth the effort to update it with the results of one's own work unless everyone does so.

In technology designed for coordination support, incentives must be aligned with the burdens imposed by the costs of sharing. We observed that there is a natural alignment in knowledge extraction, since the receiver – the one who needs the knowledge – does most of the work. In general, technologies that "pull" have a more natural alignment than technologies that "push." We observed that misalignment can result not only in a lack of knowledge sharing, but also in information overload, as generators seek to avoid the cost of selecting and translating for specific consumers, and blast out broadcasts of de-contextualized (and thus often useless) knowledge.

6.3 Sharing is Periodic or on Demand

In most cases we observed, sharing does not happen continuously. Sharing may be periodic as in the weekly all-hands meetings, or on demand as in the discussion around field test results. This has an important consequence – there is always a time lapse between the point at which something becomes known within a team and the time it becomes known across the project. The duration may be trivial (the time it takes to send e-mail to a list) or it can be very significant (the week preceding an all-hands meeting or the months between design reviews). During this time, the new knowledge is invisible to much of the project, and work that may have already been rendered useless continues nevertheless.

Awareness technologies may address the problem of making relevant work visible across team boundaries e.g., [20], but have been applied in a limited way in the engineering context. Such technologies are only useful, however, when the information they provide has a relatively high ratio of signal to noise, e.g. displaying easily intelligible and relevant information and relatively little else. While recent advances have improved our ability to detect task dependencies [10] and present such information usefully [38], more work needs to be done in order to provide the possibility of more continuous awareness at scales beyond the work team in contexts such as engineering and design.

7. ACKNOWLEDGEMENTS

Thank you to the robotics engineers for allowing us to observe your work and interview you. Thank you also to research assistants Nooreen Dabbish, Lauren Geisey, and Rohit Kabra.

8. REFERENCES

[1] Adler, P. S. 1995. Interdepartmental interdependence and coordination: The case of the design/manufacturing interface. *Organization Science, 6* (2): 147–167.

[2] Banker, R.D., Bardhan, I., & Asdemir, O. 2006. Understanding the impact of collaboration software on product design and development. *Information Systems Research*, 17 (4): 352-373.

[3] Bechky, B.A. 2003. Sharing meaning across occupational communities: The transformation of understanding on a production floor. *Organization Science, 14* (3): 312-330.

[4] Bergman, M., Lyytinen, K., Mark, G. 2007. Boundary objects in design: An ecological view of design artifacts. *Journal of the AIS, 8*(11): 546-568.

[5] Brown, J.S., & Duguid, P. 1998. Organizing knowledge. *California Management Review, 40* (3): 90-111.

[6] Brown, S., & Eisenhardt, K. M. 1995. Product development: Past research, present findings, and future directions. *Academy of Management Review, 20* (2): 343-378.

[7] Carlile, P.R. 2002. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science***,** 14 *(*4): 442-455.

[8] Carlile, P.R. 2004. Transferring, translating, and transforming: An integrative framework for managing

knowledge across boundaries. *Organization Science,* 15(5): 555-568.

[9] Carlile, P.R., & Rebentisch, E.S. 2003. Into the black box: The knowledge transformation cycle. *Management Science,* 49(9): 1180-1195.

[10] Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., & Carley, K. M. 2006. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. *Proceedings of CSCW 2006*, ACM Press: 353-362.

[11] Christensen, L.R. 2008. The logic of practices of stigmergy: Representational artifacts in architectural design. *Proceedings of CSCW 2008,* ACM Press: 559-568.

[12] Daft, R.L. & Lengel, R.H. 1986. Organizational information requirement, media richness, and structural determinants. *Management Science,* 32(5): 554-571.

[13] Faraj, S., & Sproull, L. 2000. Coordinating expertise in software development teams. *Management Science,* 46 (12): 1554-1568.

[14] Faraj, S., & Xiao, Y. 2006. Coordination in fast-response organizations, *Management Science,* 52(8): 1155-1169.

[15] Galbraith, J. 1977. *Organization Design*, Reading, MA: Addison-Wesley Publishing Company.

[16] Gittell, J.H. 2002. Coordinating mechanisms in care provider groups: Relational coordination as a mediator and input uncertainty as a moderator of performance effects. *Management Science,* 48 (11): 1408-1426.

[17] Google Lunar X PRIZE. http://www.googlelunarxprize.org/.

[18] Grinter, R.E., Herbsleb, J.D., & Perry, D.E. 1999. The geography of coordination: Dealing with distance in R&D work. *Proceedings of GROUP 1999*: ACM Press: 306-315.

[19] Grudin, J. 1988. Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. *Proceedings of CSCW 1988*, ACM Press: 85-93.

[20] Gutwin, C., Penner, R., & Schneider, K. 2004. Group awareness in distributed software development. *Proceedings of CSCW 2004*, ACM Press: 72-81.

[21] Hauptman, O., & Hirji, K.K. 1991. The influence of process concurrency on project outcomes in product development: An empirical study of cross-functional teams**.** *IEEE Trans. Engineering Management,* 433: 153–164.

[22] Henderson, R.M., & Clark, K.B. 1990. Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35: 9-30.

[23] Herbsleb, J.D., & Grinter, R.E. 1999. Splitting the organization and integrating the code: Conway's law revisited. *Proceedings of ICSE 1999*, IEEE Press: 85-95.

[24] Hinds, P., & Kiesler, S. 1995. Communication across boundaries: Work, structure, and use of communication technologies in a large organization. *Organization Science, 6* (4): 373–393.

[25] Hoegl M., Weinkauf K., & Gemuenden H. G., 2004. Interteam coordination, project commitment, and teamwork in multiteam R&D projects: A longitudinal study. *Organization Science,* 15(1): 38-55.

[26] Kazanjian, R. K., Drazin, R., & Glynn, M.A. 2000. Creativity and technological learning: The roles of organization architecture and crisis in large-scale projects. *Journal of Engineering Technology Management,* 17: 273–298.

[27] Kellogg, K.C., Orlikowski, W.J. & Yates, J. 2006. Life in the Trading Zone: Structuring Coordination Across Boundaries in Postbureaucratic Organizations. *Organization Science,* 17(1): 22-44.

[28] Lawrence, P. R., & Lorsch, J.W. 1967. Differentiation and integration in complex organizations. *Administrative Science Quarterly,* 12: 1–47.

[29] Lee, C.P. 2007. Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work. *CSCW, 16* (3): 307-339.

[30] Majchrzak, A., Jarvenpaa, S., & Hollingshead, A.B. 2007. Coordinating Expertise among Emergent Groups Responding to Disasters. *Organization Science,* 18(1): 147-161.

[31] Malone, T., & Crowston, K. 1994. The interdisciplinary study of coordination. *ACM Computing Surveys,* 26(1): 87-119.

[32] March, J.G., & Simon, H.A. 1958. *Organizations*, New York: John Wiley & Sons.

[33] Mark, G. 2002. Extreme collaboration. *Communications of the ACM,* 45(6): 89-93.

[34] Mihm, J., Loch, C., & Huchzermeier, A. 2003. Problem-solving oscillations in complex engineering projects. *Management Science,* 46 (3): 733-750.

[35] Nonaka, I., & Takeuchi, I. 1995. *The Knowledge-Creating Organization*. Oxford, U.K.: Oxford Press.

[36] Okhuysen, G.A., & Bechky, B.A. 2009. Coordination in Organizations: An Integrative Perspective. *The Academy of Management Annals,* 3(1): 463-502.

[37] Orlikowski, W. 1992. Learning from Notes. *Proceedings of CSCW 1992*, ACM Press: 362-369.

[38] Sarma, A., Redmiles, D., & van der Hoek, A. 2008. Empirical evidence of the benefits of workspace awareness in software configuration management. *Proceedings of FSE 2008*, IEEE Press: 113-123.

[39] Star, S.L., & Griesemer, J.R. 1989. Institutional ecology, 'Translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science,* 19 (3): 387-420.

[40] Teasley, S., Covi, L., Krishnan, M.S., & Olson, J.S. 2000. How does radical collocation help a team succeed? *Proceedings of CSCW 2000*, ACM Press: 339-346.

[41] Thompson, J.D. 1967. *Organizations in action*. New York: McGraw-Hill Book Company.

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