User-generated Innovation:
A Co-adaptive Phenomenon

Wendy E. Mackay
Rank Xerox EuroPARC
mackay@europarc.xerox.com

Draft: Do not cite

Abstract

The formal study of human-computer interaction is relatively new. The field is multidisciplinary and draws from a number of different research disciplines. Two basic streams of research have emerged: One treats technology as a static, independent variable, which influences the behavior of the people in the organization. The other treats the organization as the independent variable, in which decision-makers in an organization make strategic choices about technology and appropriate it for their own purposes. Neither view by itself accounts for the data reported in this paper.

I conducted a two-year field study of the introduction of a software application, the Information Lens, into an organization. Users were influenced by the software itself and a variety of other factors, which caused them to adapt their behavior. However, some users were also proactive and actively adapted the software for their own purposes. These user-generated innovations created new patterns of use which were shared within the organization. They also provided important insights about the technology and resulted in changes in future versions of the software.

The combination of these two factors, in which users both adapt to a new technology and adapt it for their own purposes, comprises a *co-adaptive* phenomenon. Neither of the research strategies described above account for co-adaptive phenomena, in particular, instances of user innovation. While it is important to build on previous, successful research strategies, it is also important that we ensure that the phenomena being studied share the same underlying characteristics. This paper demonstrates that human-computer interaction is a co-adaptive phenomenon requiring enhanced theories and research methodologies in order to adequately account for the data and make progress in the field.

1 Introduction

The computer industry has changed dramatically in the past decade. The economics of computing, in which technology has become less expensive and labor has become more expensive, has encouraged a shift in the research and the emergence of the field of human-computer interaction. The field is multi-disciplinary and is dedicated to two primary goals: to better understand the use of the technology and to influence the design of technology to increase its effective use. HCI researchers[1] were originally trained in a variety of research disciplines, with Psychology, Computer Science and Social Science as the most common, and organization theory and management science as the most common interdisciplinary specialties. Understandably, these researchers draw theories and research methodologies from their parent disciplines, modified to address the questions they find interesting in the field of HCI.

I do not want to argue against borrowing from older, more established research disciplines. We can and should benefit from knowledge and experience gained in other fields. However, there is a danger. All research paradigms rest on assumptions about the nature of the phenomena being studied. If we borrow research strategies without ensuring that the phenomena being studied have the same characteristics, we can make serious errors.

The HCI literature includes two basic research streams which are based on assumptions about the nature of human-computer interaction. One treats technology as a static, independent variable, which influences the behavior of the people in the organization. This approach leads to experiments and field studies in which a new technology is 'tested' on a set of users. The other research stream treats the organization as the independent variable, in which decision-makers in an organization make strategic choices about technology and appropriate it for their own purposes. This approach is more likely to use ethnographic methods to identify the ways in which organizations manage technology. Unfortunately, much of the work is contradictory and it is difficult to compare findings across studies.

A number of researchers have attempted to articulate the problems associated with this diversity in research strategies (Kling and Iacono, 1984, Markus and Robey, 1988, Orlikowski, 1989). Kling (1980) concentrated on the *substance* of theory, such as concept definition and normative orientation. He identified and defined six theoretical perspectives in social analyses and empirical research on computing, which differ in their definitions of technology and social setting, theoretical constructs, evaluations of "good" technology and ideologies about the workplace. Markus and Robey (1988) analyzed the *structure* of various theories, based on the theorists' assumptions about the nature and direction of causal influence. They identified three forms of causal agency: the technological imperative (in which external forces cause change), the organizational imperative (in which people act purposefully to accomplish intended objectives), and the emergent perspective (in which change emerges from the interaction of people and events).

An example of the "technological imperative" is Sproull and Kiesler's (1986) work on the impact of electronic mail on communication within organizations. They argue that electronic mail does not simply

[1]Note that I am referring to researchers here. Some of the people involved in the design and development of products may also have training in these fields, but areas such as graphic or industrial design and human factors are also prevalent.

speed up the exchange of information but leads to the exchange of new information as well. They found that decreasing social context cues had substantial deregulating effects on communication and that much of the information conveyed through electronic mail would not have been conveyed through another medium. Eveland and Bikson (1988) take a slightly different approach. They are interested in the ways in which electronic media influence the structure of work groups and their patterns of interaction. They conducted a field experiment in which they randomly assigned group members to either a computer-based or a traditional support group and gave each group the same task. They then examined the differences in communication patterns and volume of communication in general. In each case, the analysis is on the effect of the technology on the people who use it. They do not look for evidence of how the users appropriate the technology for their own purposes.

Most of the researchers who advocate iterative design strategies (e.g. Card et al., 1983, Gould & Lewis, 1985, Norman & Draper, 1986) use the "technological imperative" perspective. Some iterative design approaches (e.g. Chapanis, 1969) concentrate on setting up usability studies in a laboratory and ask users to try successive software prototypes. Users' reactions influence subsequent versions of the software. Summative evaluation (e.g. Dick, 1977) measures reactions to the finished software and is often used by software manufacturers to ensure that the software is of sufficiently high quality to ship. Participatory design (e.g. Ehn, 1989, Greenbaum & Kyng, 1991) is another approach, in which software developers and researchers work together with users, who actively provide input into the software design. These strategies are all designed to improve software by making it easier for users to explore and learn, increasing user efficiency, and either reducing the likelihood of errors or to making breakdowns more easily repairable (Brown and Van Lehn, 1980). However, they all treat the users as relatively passive recipients of the technology; the goal is to improve the technology so that the impact on users is improved rather than to create technology that users can better appropriate for their own purposes.

The organizational behavior and information technology literatures are more likely to adopt the the "organizational imperative" research strategy. For example, Zuboff (1988) examines how the choices of management can fundamentally affect the use of technology in an organization. She argues that technology can be designed to "automate" work, treating users as simply another component in a work process, or to "informate", empowering users to make decisions and operate autonomously. She feels that decision-makers in organizations should choose technology that empowers users, both for economic reasons, because users will be more productive and better able to handle unforeseen circumstances, and for humanitarian reasons. Yet Zuboff's analysis does not take into account the level to which individual users appropriate technology and make choices about use. They *react* to information, which changes their behavior and thus the behavior of the system. More importantly, they are sometimes *proactive*: they look for information from the system. They bring assumptions and experience and apply them to the technology in both anticipated and unanticipated ways. They may actively pursue information from the system, even if the system is not designed to provide it. Zuboff's point is that we can design software to encourage these activities, but it is important to note that some individuals will do it anyway, even without encouragement.

Paragraph with more examples of organizational imperative studies. Bailyn? See Barley.

Individual members of an organization are influenced both by the technology and by organizational factors. Individuals are influenced by how others use the system (either members of their group, friends, or people who are geographically nearby). These activities change the organizational context and group norms of behavior. Research methods that treat the organization as the independent variable and the software technology as a dependent variable are inappropriate, because they do not sufficiently account for the influence of the software design and individual actions.

The third approach, the "emergent perspective" is least common. Barley's (1986) study of the introduction of CT scanners into two radiology departments is a good illustration. Barley found that the introduction of the same technology into two different organizations resulted in different organizational structures. He discusses how the organizational structure is changed by the introduction of new technology. However, this analysis does not examine the effects of technology on individuals in any detail nor on how individuals affect the technology.

Suchman's (1987) work could also be framed as an emergent perspective, but in a somewhat different way. She criticizes the dominant assumptions of cognitive science and artificial intelligence, which assume that people "plan" their interactions with technology. She argues that it is essential to account for the "situatedness" of human social behavior attempting to understand how users interact with technology. She uses the term *situated action*, which "underscores the view that every course of action depends in essential ways upon its material and social circumstances. Rather than attempting to abstract action away from circumstances and represent it as a rational plan, the approach is to study how people use their circumstances to achieve intelligent action. Rather than build a theory of action out of a theory of plans, the aim is to investigate how people produce and find evidence for plans in the course of situated action." (Suchman, 1987, p.50) The implication for design is that artifacts by themselves do not determine their use, but rather the concrete circumstances of users' encounters with these artifacts determine use.

Orlikowski (1989) also identifies two streams of technology research, which she describes as the "technological imperative" model and the "strategic choice" model. Orlikowski proposes an alternative framework, after Giddens's (1984) concept of structuration, an "attempt to explicate how social practices are produced and reproduced across time and space...information technology can be seen to operate recursively: while on the one hand information technology is a product of social practices, it also facilitates further social practices (reproducing or even changing them)".

The interaction between human agents and technology is defined by two relationships: technology as a product of human action and technology as a medium of human action. When viewed as a product of human action, technology is seen to be a human artifact, built within certain social and historical circumstances. It only has an influence because it is appropriated by people in the execution of certain tasks. When viewed as a medium of human action, it is seen to both facilitate and constrain the performance of tasks. The extent to which either dominates depends on the "intentions of the designers and implementors, the institutional context in which the technology is embedded, and the autonomy and power of the technology users."

In addition, human action is situated action, shaped by the organizational context. Finally, peoples' uses of technology may operate directly on the institutional structure of an organization, either by reinforcing it or transforming it. According to Orlikowski, "These effects -- comprising the *unintended consequences of interaction with technology* -- are often not reflected on by users, who are generally unaware of their role in either reaffirming or disrupting an institutional status quo."

The structurational model makes a number of observations:

1. Technology is a human artifact, built within certain social and historical circumstances.
2. Technology only has an influence because it is appropriated by human action in executing certain tasks.
3. Technology can condition but not determine social practices: human agency is required.
4. Technology is both facilitating and constraining, depending on organizational conditions.
5. Human action is situated action, shaped by organizational contexts.
6. Human action, through use of technology, acts upon the institutional structure of an organization, either reinforcing it or transforming it.

In order to study these questions, it is important to examine the behavior of individuals within an organization over a period of time. Isolated behavior within a laboratory setting would not provide useful answers to these questions; nor would a field study that concentrated on the organization's use of the technology. It is important to study individuals, to identify both common patterns of behavior and the diversity in their interactions with the software, within an organizational context. The following study demonstrates how users both adapt to the technology as it is presented to them and then adapt it for their own purposes.

2 Research Study

I conducted a two-year study of how users in a particular organization manage their electronic mail and the effect of introducing a software application, the Information Lens (Malone et al., 1987a), on their mail handling processes. I wanted to observe how people actually use Lens "in the field", in the course of their daily work. This study was conducted in three phases, which correspond to three versions of the software. This was done for both logistical reasons and because the group of study participants changed with each new version of the software.

2.1 Software: The Information Lens

The Information Lens is designed to help users filter and organize electronic mail and thus increase the effectiveness of their electronic communication. Lens accomplishes this by filtering *out* uninteresting messages (e.g. users can process high-priority messages first and delete low-priority messages), and filtering *in* interesting messages, (which allows users to receive messages that would not normally be accessible). The goal is to increase the effectiveness of mail handling by increasing the proportion of useful messages. Users can create their own sets of IF-THEN rules; incoming messages are then processed according to those rules. The rules usually search for text strings in the TO:, CC:, FROM:, and SUBJECT: fields and perform various operations, such as deleting the message or moving it to a specified mail folder.

The design and theory behind the Information Lens has been discussed extensively elsewhere (Malone et al., 1987a, Malone et al., 1987b), as has a newer system called Object Lens (Lai and Malone, 1988), which generalizes and extends the ideas from the Information Lens. Many software manufacturers have expressed an interest in the ideas and variations of Lens are being added to electronic mail products. I have reported about the use of electronic mail (Mackay 1988) and Lens (Mackay et al. 1989, Mackay, 1989a) elsewhere; the next section describes the research study from which these data were collected.

2.2 Research Site

The research site was a large laboratory (approximately 60 people) in a research center for a multi-billion dollar international corporation. It was chosen because it met two key criteria: a) the hardware and software environment necessary to run the Lens prototype was widely available, and b) the members of the organization were experienced users of electronic mail who were currently facing the information management problems that the Information Lens was designed to address.

People at this site had been using electronic mail extensively for over a decade and a half. Essentially all employees at this site relied on electronic mail for both formal and informal communication; people received an average of 30 to 35 electronic messages per day. Many of these messages came from a variety of distribution lists on both work and non-work topics. Most members of the lab, including managers and secretaries, had workstations on their desks which were linked together in a distributed network that permitted easy exchange of files and electronic mail messages.

Users interacted with the software with a mouse and keyboard and multiple applications could be accessed simultaneously in different windows on the screen. Users usually chose applications via menus, although a command language was also available for access to applications that were used less commonly. Everyone used the Lafite mail system, which has a graphical interface for composing, reading and browsing messages and mail folders. Messages could be automatically marked with system-defined characteristics, such as *moved* or *deleted*, or with characteristics specified by the user. Lafite uses the concept of message templates, which users create when they need to repeatedly compose similar types of messages, e.g. bug reports and visitor logs. Users could create and name as many mail folders as they liked. The Lens prototype was designed to enhance rather than replace this system; the goal was to allow users to chose any or none of the new Lens features, as desired.

Pilot Phase: The first prototype of the Information Lens was developed at MIT and was designed to work with the Lafite mail system. The prototype was used by a small group of software developers in the research lab at MIT. I interviewed the original developers about their use of Lens. The results contributed to the subsequent design of the interview and automatic data-collection procedures that were used at the research site. When the prototype of the Lens software was deemed sufficiently stable, an external research site was chosen to test the software.

Phase One: The prototype was brought to the research site and I conducted preliminary interviews at that time. Due to various bugs and other problems, all but one person abandoned the software after several days. I interviewed the users about their experiences with the software shortly after this time.

Phase Two: Based on the experience with version one, the software was revised at MIT over a period of six months. In order to better understand and accommodate the needs of the users at the research site, the final set of changes were made by an MIT programmer who spent the summer there. I visited approximately one month prior to the planned introduction of version two to collect baseline data about use of electronic mail. Version two was introduced eight months later. The entire lab was invited to a talk about Lens and a later "kick-off" meeting with a live demonstration of the software. Members of the lab were informed of these events and invited to participate in the study via electronic mail. 18 people volunteered to try Lens. I helped each person install the software and create a first rule. I conducted two sets of follow-up interviews at one-month and three-month intervals.

Phase Three: Version three was created by a local programmer who had actively assisted with the first two phases of the user study and had created a simple version for his own use. He then created a more substantial version, written in Common LISP and Common LOOPS, that could be used by others in the organization. This version was created to work with the new version of the Lafite mail system and the new version of the LISP system environment. Version three was smaller, easier to maintain, and incorporated the features most requested by users. It became the most popular version of the software and drew the most users, even though it lacked some of the features of the original Lens (including message types and the anyone server). Version three became generally available six months after version two and support for the former was terminated. Since Lens users were only able to convert to version three when their workstations received the new LISP software upgrade, start dates for version three ranged over a period of several months. I conducted two sets of follow-up interviews three months and six months after version three was first introduced.

2.3 Participants

The lab is composed of research scientists, software engineers, managers, secretaries and other administrative personnel. Many members of the technical staff have Ph.D.'s in a variety of subjects, including Physics, Psychology, Computer Science, Anthropology, Mathematics and Sociology. The lab is divided into small groups which are responsible for one or more projects (some with outside funding). Each working group has a manager, secretary, one or more senior technical staff members and usually several junior-level technical staff. Many of the latter are students from local universities. The members of this laboratory are more highly educated and accustomed to more advanced computer technology than a representative sample of office workers. However, the participants in the study do include a range of job categories, educational backgrounds, and levels of computer experience.

Participation in the study was voluntary and different people participated in different phases; altogether, 26 people participated. The study was encouraged and partially supported by the head of the lab. The person who developed the local version of Lens was omitted from the analyses because many of his rules were created to test the system and could not be distinguished from his regular mail processing rules. In order to disguise the identities of the participants, their names and some of their personal characteristics have been changed.

Phase One: Six members of a small working group, including the manager, were asked to try version one of the software. All of these people were currently or formerly programmers.

Phase Two: Participation in phase two was solicited via an electronic mail announcement to the members of the lab. This ensured participation by active users of electronic mail. The message described the prototype of the Information Lens and asked for volunteers. 23 people, including two people from phase one, expressed an interest in trying version two. Five of these people ultimately chose not to use Lens, either because they were reluctant to try prototype software or because they did not trust Lens to handle their mail before they had seen it. Of the 18 who decided to try Lens, three were managers and 15 were full-time researchers. Of these, six were computer scientists and nine were trained in Physics, Psychology, Anthropology, or Sociology. The five who chose not to use Lens included three administrators, one computer scientist and one manager.

Phase Three: Members of the lab were told about the availability of version three via electronic mail. 26 people either converted to or began with this version of the software. Over half of these people had previous experience with an earlier version of Lens. Mackay et al. (1989) reported how a subset of this group, "experienced users" who had used version three for three to seven months, used rules. The 13 participants included a manager, a manager/secretary team[2], four computer programmers, one research scientist with formal training in computer science and six other research scientists (from a variety of physical and social sciences) without formal training in computer science. The people omitted from this analysis had either stopped using Lens or had not yet accumulated three months of experience with version three of Lens. The most common reason given for not continuing use of Lens was use of a workstation that did not support the new Lens software. Two people (one programmer and one manager) reported that they did not find Lens sufficiently useful.

---

[2]The secretary used Lens for both her personal mail and her manager's mail.

2.4 Data

Two kinds of data were collected: participant interviews and automatically-collected records of use of the Lens software.

2.4.1 Interviews

I conducted three types of interviews. Preliminary interviews were designed to identify existing patterns of mail use, prior to using Lens. Installation interviews were designed to help the user install and begin using Lens, as well as provide a second look at existing mail use patterns. Follow-up interviews were conducted at varying intervals after the user had been using Lens and were designed to look at how users used Lens and its effect on their mail handling strategies.

All interviews were scheduled for one hour in the participant's office. Prior to each interview, I asked the participant to save one day of mail messages, deleting any that were confidential. In the preliminary interviews, I asked questions about the user's background, including programming experience and job responsibilities. In all interviews, I asked a series of questions about the person's use of electronic mail. I asked them to estimate the daily numbers of messages sent and received, the number of mail folders, the size of the inbox and the number of distribution list subscriptions. These answers were checked against the actual numbers for the current day and participants were asked if the day was typical. I also asked open-ended questions about major problems and successes with electronic mail. Participants used this an opportunity to describe their current communication patterns, successful mail management strategies, problems that needed to be addressed and whether or not particular Lens features would be expected to help.

In the installation interviews, I helped users install Lens and dealt with any system problems. During the installation time (approximately thirty minutes), I repeated the questions about electronic mail use from the preliminary interviews and asked additional open-ended questions. I gave each participant a five-page mini-manual, which explained the key Lens features on one page, followed by a three-page description of how someone might use it on a typical day, and a page with five sample rules. Those who requested it were also given a 45-page user manual (Mackay, 1986b). I helped each user chose a first rule, create it with the rule editor, create a test message, mail it, and check to see that the rule fired. Most people chose a variation of one of the rules suggested in the mini-manual. Users were responsible for creating additional rules by themselves.

In the follow-up interviews, I repeated the questions about mail use and then asked about their use of Lens, current choice of rules and what effects, if any, Lens had had on their general mail handling strategies. Interviews were conducted over a number of visits, shown in Table 1. The last three visits in phase two included a combination of preliminary, installation and follow-up interviews.

Phase One: I conducted one set of preliminary interviews and a set of follow-up interviews shortly after the participants tried to use version one. Because all but one person had stopped using it, I did not include many questions about use of Lens. I did, however, gather many suggestions for improving the software. One person continued to use Lens; he eventually developed the local version of Lens that was used in phase three of the study.

Table 1: Dates of interviews at research site

Phase	Start		End		Year	
1	10/13	Mon	10/14	Tues	1	
1	1/27	Tues	1/29	Thur	2	
2	7/20	Mon	7/24	Fri	2	
2	8/20	Wed	8/26	Wed	2	Kickoff
2	9/1	Tues	9/4	Fri	2	
2	11/5	Thur	11/6	Fri	2	
3	4/14	Thur	4/19	Tues	3	
3	7/13	Wed	7/15	Fri	3	

Phase Two: Most of the preliminary interviews were conducted one month prior to the introduction of version two. I also conducted several preliminary interviews for the people who volunteered to participate during the kickoff meeting. Installation interviews were scheduled immediately after the kickoff meeting. I also conducted a few installation interviews when I returned the following month for follow-up interviews. I conducted two sets of follow-up interviews at one-month and three-month intervals after the kickoff meeting.

Phase Three: I conducted one set of interviews three months after the conversion to version three. On this visit, I also asked participants to rate the lens features on a scale of 1-5. I conducted another set of interviews three months later. I noted the effects of converting from version two to version three of the software, which required users to recreate their rules. Most of the interviews in phase three were conducted as follow-up interviews, because most of the participants in phase three had been involved in phases one or two. Several people started using version three of Lens without joining the study; I did not interview these people.

2.4.2 Automatic data collection

The automatically-collected data were recorded for a period of over two years. This analysis will concentrate on data collected in phase two and the first seven months of phase three. These data were recorded in LISP files (called "Lens profiles") and include the user’s rules, Lens customization preferences, and the hierarchy of mail folders. Later versions were enhanced to record when each rule was created, changed, deleted or fired. A separate program translated the LISP file into a more easily-readable form. These data were collected weekly or every time the user changed any aspect of Lens. In addition, I also collected data about the user's distribution lists within the company. Members of the lab were placed on some of these distribution lists automatically and joined others voluntarily. (I did not have access to external distribution lists.)

3 Results

The standard way to present the results of this study would be to identify the ways in which users reacted to the various software features in Lens. Studies of this kind have already been reported, including an analysis of the organization's use of electronic mail (Mackay, 1988, Mackay, 1989a) and how experienced Lens users used the software (Mackay et al., 1989). This approach is useful if one is interested in finding out how particular features of the technology are used and whether or not they are useful.

However, this type of analysis provides a very limited view of the use of the technology within the organization. Even referring to the software as "the Information Lens" is an oversimplification. Does "Lens" refer to the system described in various research articles? Is it the original prototype? Is it the prototype that was most widely used? What about the Lens spin-offs, Object Lens (Lai and Malone, 1988) and Argus (Mackay et al. 1989b) , which generalize and extend different sets of concepts from the original Information Lens? Clearly, any or all of these could be considered the Information Lens.

In order to understand the use of Lens, we first need to view the software as a collection of ideas, expressed through several implementations and changed by both users and developers over a period of time. While it is sometimes convenient to talk about the Information Lens as if it were a unitary phenomenon, this practice has the unfortunate effect of implying that Lens is static. It encourages us to misunderstand the process by which the software itself is changed through design and use. During the course of this study, it became apparent that Lens was extremely dynamic. Not only did the technology itself change, through updates and new versions, but the perception of it changed, shaped by the developers, users, researchers, and even people who read about Lens technology.

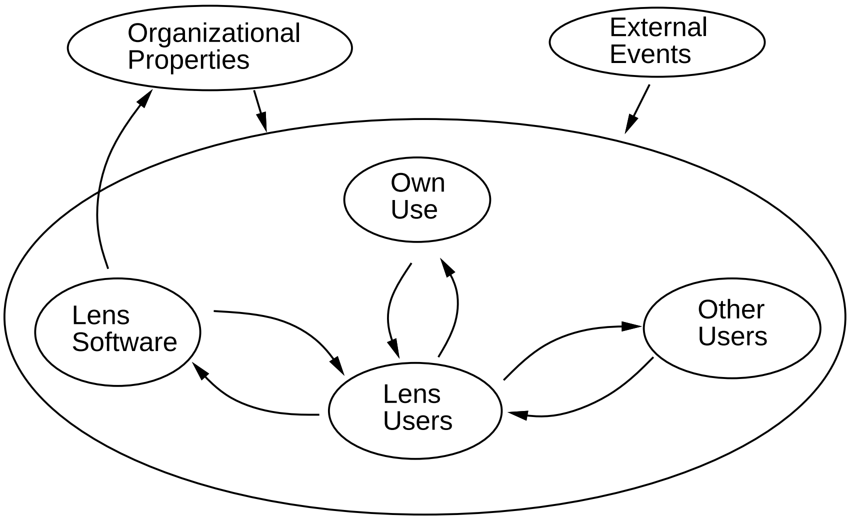

Figure 1: Influences on the use and interpretation of Lens

This results section illustrates some of the factors that changed the patterns of Lens use over time. This section is organized according to the various influences among the software, the users, and the organization, illustrated in Figure 1. Lens users are both influenced by and influence the software, their on-going use of Lens and the use by others in the organization. Various properties of the organization and external events also affect the use of the software over time. All of these elements evolved over the

-

two-year study period and changes in each influenced the others and contributed to multiple and evolving models of use of the software. Together, they comprise the complex definition of Lens as it is used by this organization.

3.1 Lens users adapt to Lens based on various influences

The model of use of Lens changed significantly over the study period. In fact, it is more accurate to say that the initial model of use became an interlocking set of models adopted by individuals and groups within the organization. These models of use were influenced by various factors:

1. **Lens software**: The software itself and the model of use presented to users.
2. **Individual use**: The results of previous experience with Lens.
3. **Other users**: The experiences of other users in the organization.
4. **Organizational properties**: The norms of behavior and expectations in the organization.
5. **External events**: The unexpected events that change people's daily routine.

3.1.1 Users are influenced by the Lens software

The Lens software was formally introduced to the members of this organization through presentations, individual discussions and demonstrations. The eventual users were clearly influenced by this; simply discussing the concept of automatic sorting of messages caused them to examine their own mail processing strategies, sometimes for the first time. They attempted to articulate their individual strategies, based on perceptions of their mail volume and their strategies for handling mail. The Lens demonstrations provided scenarios for solving mail problems, which led people to develop expectations about Lens and how it might help them. These perceptions and expectations formed the basis for their predictions about how they would use Lens and served as the basis for the first model of use: an "automatic secretary".

The automatic secretary model was intended to pre-sort mail into prioritized groups of messages. Some people liked the idea: *"My goal is to read as little [mail] as possible."* Others didn't: *"I don't want automatic mail sorting because I don't trust a formula for sorting -- I'm afraid it would get sorted and I wouldn't look at it again."* and *"I could never trust a machine to read my mail for me."* (Note that new managers often have the same reaction to giving up control over their paper mail to their new secretaries.)

Lens includes more than just the advertised features of rules and message types. Each implementation provided a number of additional features and characteristics that affected users in different ways. An example of a well-liked feature was a direct-manipulation interface for managing a hierarchy of mail folders, added to version two by the developer of version three. Users were able to transfer their folders to this format, and then rearrange folders to different positions in the hierarchy. As one person said: *"I love the folder tree, I use it all the time...it encourages you to have different types of folders."* Of the 23 people who tried it in different versions of the software, all but three reorganized their mail folders from a single level to a tree structure with two to four levels. This affected their use of folders: *"I created more* *folders than I would have -- it comes in handy when I want to separate out [subcategories]. Otherwise they get too big...It makes everything easier to find."* However, two people who tried it later decided they didn't like it: *"I generated a gorgeous tree of about 25 folders, but I only use 4-5...I don't want my mail to be that complicated."* Two others wished the folder tree took less space on the screen.

The presence of new options in version three affected users' perceptions of Lens and thus their decisions about how and when to use it. For example, the addition of multiple rulesets to version three caused eight out of ten people to modify their strategies for processing their mail. One person switched completely from one strategy to another. These examples demonstrate different ways in which people were affected by the changing Lens technology. Some were affected simply by the discussions of Lens. Others used various Lens features and subsequently changed their strategies for handling mail.

3.1.2 Users are influenced by their own use

Individuals were also influenced by discoveries made about their own use of the software. In most cases, user's predictions about how they would use it differed from their subsequent use. Perceptions of Lens evolved over time, influenced by experience and changes in mail handling strategies. Most people did not evaluate their use systematically; only three people mentioned that they used their Lens profiles to evaluate the effectiveness of their rules. One user described how her views had changed through her own explorations: *"Sorting seemed most useful at first, then you poke around and discover things."*

Many people reported changes in their use of Lens, their resulting behavior and in their perceptions of what they do. Some reported changes in their mail reading habits:

> *I don't read all my mail [anymore], I have a junk folder. I am pickier about what I read now.*
>
> *I read mail less often than I did before.*
>
> *I used to read, then sort. Now, I sort directly.*

Others began to attend more lab meetings:

> *I have attended more [lab meetings]. Without Lens, I probably would have deleted those messages.*
>
> *I now go to [lab] meetings that I would have missed before.*
>
> *I read all [research site] messages more carefully and attend more [lab meetings].*

Some people developed new strategies for dealing with distribution lists. Three people reported that Lens made it easier to stay on distribution lists. One said:

> *I don't remove myself from distribution lists, I just auto delete the messages. Occasionally, I spot something in the deleted messages and take a look. I'm always disappointed and say to myself, 'Yes, I should have deleted it'. It's easier to change a rule than get off the d.l.[3]"*

---

[3]A "d.l." is a distribution list.

For others, Lens gave them feedback that enabled them to get *off* distribution lists:

> *I dropped off several d.l.'s.*
>
> *[Lens] made me realize I was getting mail from useless d.l.'s.*

Several people discovered that they did not have the self-discipline to look into folders with important messages, if they had been sorted there automatically. One person described his experience: *"I used to automatically move project-related messages into another folder, but then I didn't check it. So now, if it's important, it stays in 'active'...it's the calendar I look at every day. Most important messages stay in active. Otherwise, it's 'out of sight, out of mind'."* These changes in perception of their own behavior affected the user's subsequent use of the software.

3.1.3 Users are influenced by other users

Lens users suffered from a common complaint: busy people object to learning about new technology: *"My basic reason for not using Lens is that it's 'yet another widget'...I generally have a very conservative approach to learning new packages unless they're very useful... They're just that much more to learn and to screw up."* These users rarely take the time to figure out how to adapt the technology to meet their current needs: *"I didn't write any rules...[because I] couldn't think of anything useful."* This resistance to spend much time learning something new lead people to ask for examples and learn from the experiences of others: *"I'm busy and have a low tolerance for learning new things. I'd like a profile written by a Lens wizard."*

Asking for help from others has an important effect on the overall organization: a small number of people create patterns of using the software and others adopt those patterns. Choices about which pattern to use were influenced by the user's existing patterns of mail use and perceptions about Lens. By the end of the study, three basic patterns of use had emerged, which influenced the ways in which all users created their rules:

1. *Prioritizing:* The 'automatic secretary' model: Rules that either move important messages out of the inbox into a "priority" folder or move unimportant messages out of the inbox, leaving the inbox as the priority folder.
2. *Archiving:* Rules that sort messages after they have been read, to facilitate subsequent retrieval of technical or other information: *"I want to see it first, then archive it."*
3. *Context-specific:* Rules that are run on specific occasions, e.g. returning from a trip: *"When I'm back from a trip, I get tough. I'm vicious about messages... I wipe out seminars, etc."*

When the ability to create multiple rulesets appeared, people talked to each other and decided to try each other's strategies. Two people discovered ways to copy all of another person's rules. In one case, the rules did not prove useful and he stopped using Lens. The exception was when he returned from a long trip and used the rules to successfully filter the mail that had accumulated. This encouraged him to start using version three of Lens. He created a completely new set of rules and then "borrowed" specific rules from the person he'd copied from before. The second person who copied an entire set of rules later edited them to be more useful. He recreated the rules somewhat differently when he converted to version three, after he had obtained some experience using the rules.

Although some individuals in the organization created their own unique sets of rules, many shared rules with each other. Borrowing rules has numerous advantages for individual users. They can reduce learning time and lower the risk of making errors, which increases the time available for accomplishing actual work. They can also experience how other people work, find out out new ways of doing things and benefit from each other's innovations. Four people became the sources of these rules:

Researcher	Announced the Lens study, interviewed participants and provided a set of five example rules when they got started.
Developer	Created the local version of the software, based on own ideas and findings from user's experience with the earlier version of Lens.
"Regular user"	Non-technical person who liked and used the locally-developed version of Lens. Shared with other non-technical users.
"Lens wizard"	Highly technical, one of the first users of the first version of Lens. Immediately wrote almost 100 rules, provided feedback and ideas to the developer of version three.

As the researcher, I provided the most visible source of rules and rule examples. I did not provide any rules in phase one. However, in phase two, I selected five rules that illustrated the strategies that the participants in the previous study had found useful. Most of the participants in the study tried at least one of these rules.

The programmer who created version three of Lens gave advice about rules to approximately one third of the active Lens users, but did not provide direct copies for them to use. He introduced a number of innovations, which were incorporated into the design of the software and made available to everyone. One might have expected him to be the primary source of sharing rules, because of his knowledge of the software. However, he was more likely to provide advice and support than sample rules. He did not share his own rules, which were designed to test the system and were quite different from those of a normal user.

The "regular user" was a non-technical person who liked and used versions two and three of Lens. He adopted a prioritizing strategy from a colleague and then created his own, very stable, set of rules. Another manager in the group then borrowed his complete set of rules. This manager felt that they had similar mail needs and that the shared rules would provide a simple, useful model for her to get started with.

The "Lens wizard" was the earliest and most steadfast user of Lens. Considered one of the top technical members of the lab, he immediately began experimenting and created almost 100 rules. When asked why he experimented so much, he said: *"Lens gets more valuable as more people use it...so somebody has to start."* Throughout the project, he conferred with the developers and made numerous suggestions; he was also vocal with complaints. He created a number of innovations that were incorporated into the local version of the software.

The rule wizard created a number of complex, but useful rules that could be traced in other people's rulesets. A good example is the "Boring" rule, which was borrowed by five other people. Another example is a pair of rules that identify a particular author in a commonly-read jokes distribution list. The first rule identifies and saves jokes by this author; the second identifies all other jokes and deletes them. Three people borrowed this pair of rules exactly and another adopted a variation that identifies a different type of joke ('sniglets'). Two people borrowed his rules for deleting messages from two people who violated the rules of distribution list etiquette. The function served by the rule wizard appears similar to that of the gatekeepers identified by Allen (1972). Gatekeepers are skilled individual contributors who actively seek technical information outside of the organization and translate it for the use of their colleagues. In this case, one person developed strategies for using rules, which he shared with his colleagues.

3.1.4 Users are influenced by organizational properties

Organizational properties affect the characteristics of the work and social environment in the lab. Examples include the computing environment, e.g. the use of workstations connected via a high-speed network and the use of electronic mail as an essential medium for communication. Other properties include accepted norms of behavior, e.g. the etiquette in using electronic mail. The organization also supports several kinds of routine messages: A receptionist takes phone messages and sends them to the recipient via electronic mail. Members of the lab are expected to send a "visitor's log" message whenever they host a visitor. Seminar and lab meeting announcements are sent out via electronic mail and everyone in the lab is invited. A special program allows users to send hardware and software bug reports to the appropriate person, with all the necessary system information filled in automatically. These factors influenced the people in the organization and how they used Lens.

Social factors greatly influenced people's decisions whether or not to try each different version of Lens. Members of the first study group cited three kinds of reasons for using Lens:

1. *Technical reasons*: They found the Lens talk and ideas interesting.
2. *Practical reasons*: They wanted to solve existing mail problems.
3. *Social reasons*: They felt peer and management pressure to participate.

When the first group of users found the version one prototype slow and buggy, they made their objections known and others were reluctant to try version two as a result. However, the influence of several lab managers, the presence of several support people and revisions in the software encouraged other people to give version two a chance. Version three was the most well-accepted: it was written and supported by a local, highly-respected programmer, reflected many of their comments and suggestions, provided a simpler user interface based on the local interface standard, and several other influential members of the lab had tried it and liked it.

Social factors also influenced people's decisions to continue using the technology. Several people commented that they did not like early versions of Lens because their friends reacted poorly to extra empty fields: *"[People] didn't like getting all these strange headers"* and *"the header has garbage that nobody wants to see."* Another pointed out that the *"length of the message affects readability and Lens lengthens messages."* These people didn't like the fact that they had no control over how the final message looked and reacted better to version three because *"it looks more like people generated [the message], not a machine."*

Reactions to message types were mixed. Some people simply wanted to use the existing types: *"I want the [Lafite message types], not the message type tree."* Several people commented that they would be more likely to use the message types provided in Lens if other people did: *"I avoided using message types...I don’t see any point unless the whole community uses them."* and *"I would use [message types] more if more people did."* One person felt that the message types that were provided were *"too simplistic...like talking to Ann Landers."* Another felt that his friends would think it silly if a request to borrow a book was labeled *"request for action."*

Some of the problems experienced with the early versions of Lens had to do with differences between the work environments at the development and user sites. On the surface, they were quite similar; the hardware and software platforms were the same. However, different patterns of use by individuals rendered certain characteristics of the Lens software unacceptable. For example, one manager objected to the length of time Lens took to update the Lens profile when he wanted to log out of his workstation. He had a noisy machine and a small office, so he liked to turn the workstation off when he held meetings there. Because Lens added an extra three to five minutes to the logout process, he decided to remove Lens from his system. Even though this method of saving profiles was later changed, he never resumed using Lens. The developers of Lens never experienced this problem because they tended to keep their workstations on and rarely waited for it to logout. The differences between the two work environments thus affected the perceptions of the software.

The lab has strong, though unwritten, rules of electronic mail etiquette. Many people commented unfavorably about breaches in etiquette and a few people cited etiquette violations as a key reason for using Lens. For example, the members of different groups or geographical locations are automatically placed on certain distribution lists. A common complaint was the indiscriminate use of these lists for trivial purposes, such as asking about lost keys in the parking lot. Another undesirable practice was the indiscriminate use of "cc:", copying a message to a large number of people for no apparent reason. This sometimes happened when a user replied to a message from a distribution list and the mail system automatically copied the reply to the distribution list.

Special interest distribution lists are similar to special-interest bulletin boards or news groups. People subscribe to the ones they find interesting and may forget they exist if the traffic is low. People describe these lists as having their own personalities and sets of expectations. Some generate and allow a great deal of "flaming", which involves heated exchanges over a previous message or hot topic. "Flaming" is expected and acceptable on some lists, but not others; people quickly learn that it’s a bad idea to "flame" on the staff lists. Different people tolerate different amounts of flaming. Some tolerate a lot: *"It’s worth it to see 90% and delete in order to get 10% good stuff."* Others do not: *"So many people use email without etiquette. I got off a lot of d.l.’s and am down to a very few general lists...It’s a shame."* This person also commented that you have to watch carefully, because *"good lists go bad."* Another annoyance for people who read a distribution list is the behavior of newcomers, who ask questions and raise issues that have already been resolved. *"We need a way for new people on a d.l. to go look at old questions and the etiquette for using the list."*

Another etiquette issue involves the use of network resources. "Autogetmail", introduced in phase two, automatically retrieves new mail messages for the user. Autogetmail was the highest-rated feature of Lens; one person even cited it as her reason for using Lens without rules. For some, it affected mail reading behavior: *"My mail reading pattern changed -- now I read it when I want to, not when it arrives."* Although they initially rated it highly, four programmers reported four months later that they had either stopped using it or used in under very restricted circumstances: *"I don't use autogetmail anymore because of network issues...it keeps an open connection with the server...it's a matter of network etiquette."* and *"I don't run it continuously [anymore]."*. As technical people, they were aware of the impact of their actions on others in the organization and reacted by avoiding or restricting their use of autogetmail. However, less technical people, who didn't realize the impact of using this feature, continued to use it: *"Autogetmail is really great...I would be offended if I had to get my own mail."*

Organizational properties also affect the kinds of rules that people write. A good example is the use of deletion rules. Mackay et al. (1989) evaluated a group of rules from 13 experienced users and found that deletion rules comprise a small (20%) percentage of overall rules. 60% of rules that delete messages are qualified through specification of more than one field, whereas only 32% of rules that move messages are so qualified. Although a number of people feared that Lens users would write rules to automatically delete messages from them, this was never done. Personally-addressed messages were treated as significantly more important than broadcast messages. Only five of the 38 rules that deleted messages included the sender and all of these further qualified the deletion based on the content. These rules deleted messages from people who violated accepted standards of mail use, but were outside the range of peer pressure. One person was an outsider who had discovered how to reach the site and regularly sent messages about his new-age seminars. The other person taught aerobics classes several times a week and sent messages to the entire staff whenever it was time for another lesson. Although the authors of the rules that deleted such messages did not expect personally-addressed mail from either of these people, all were careful to include both the sender's name and another field that identified these messages. A personally-addressed message from either of these people would not have been deleted. These deletion rules were actually shared rules: one person created them and other people borrowed them.

People who write rules must deal with a trade-off when deleting messages. If a message is deleted, it will either be correctly deleted (correct rejection) or incorrectly deleted (false alarm). If a message is not deleted, it will either be correctly kept (hit) or incorrectly kept (miss). People base this trade-off on a number of factors. The most important appear to be the risk associated with not responding to a message and the person's desire to keep messages. The former is greatly affected by the properties of the organization and the consequences for any individual of not responding to mail messages. Some people can afford to incorrectly delete a message: *"An accidental delete is not a problem."* For example, if a manager does not respond to a message requesting something, the sender will try again, phone or stop by. Because managers often delegate tasks via mail, it is generally up to the recipient to respond. Managers also receive a higher percentage of messages for purely political reasons, such as messages copied to the manager to increase the perceived importance by the intended recipient. Managers and high-ranking staff can thus use stricter deletion rules and only worry about reducing the number of messages that are kept incorrectly.

In contrast, people in support positions, such as secretaries and software maintainers, often conduct much of their work via mail. One secretary commented that *"most of the stress in my job comes from email."* They cannot refuse most of the tasks delegated in this way and must process a higher percentage of their mail. They must also accept a higher percentage of incorrectly-kept messages in order to lower the possibility of incorrectly deleting messages that might be important.

The organizational properties described in this section, as well as others, affect the individuals in the organization in different ways. Factors such as the individual's role the organization, accepted standards of behavior and the characteristics of the computing environment, affect the choices they make about how and when to use Lens.

3.1.5 Users are influenced by external events

Sometimes, an upcoming event causes users to change their use of the system. One user had been reasonably satisfied with Lens in phase two. His rules had been designed to sort mail after he had read it: *"Nothing happens automatically, I look, then sort."* One month, he faced an impending mail crisis: he was planning to go away on a long trip and was also responsible for an upcoming scientific conference. At the same time, version two had been "retired" and he needed to recreate his rules in version three. Because he was in a hurry, he decided not to recreate all seven of his existing rules. Instead, he created a new rule that identified messages addressed directly to him (as opposed to messages addressed to a distribution list of which he was a member). He subsequently added one more rule, which identified messages about the conference and forwarded them to his assistant. When he returned from his trip, he found that these rules had separated out most of the important messages and left the less important messages. What began as an emergency measure to solve a crisis became his standard strategy for handling mail. He said these two rules *"changed my life."* He thus converted from a post-sort (archiving) strategy to a pre-sort (prioritizing) strategy. Although he intended to recreate an additional ruleset for sorting rules, he never got around to it: *"I hate to stop long enough to get set up."* These two rules significantly changed the volume of mail he processed. Instead of scanning the headers of all messages, he could now look at a significantly smaller subset of more important messages. Note that this changed how he partitioned the task of reading his mail. Instead of reading both important and unimportant messages all at once, he could put off viewing unimportant messages until he had time. The net result was that reducing the number of rules and running them at a different time increased the overall effectiveness of his mail handling strategy.

External events appear to play an important role in influencing *when* users decide to modify their rules. Although I do not have access to all possible relevant events, several have clearly influenced users: long weekends, organizational changes, trips, changes in the software, interviews by the researcher, and the conversion from version two to version three.

Two people reported that they liked to experiment with rules when they had extra time, usually on a holiday weekend. At least two changed their rules in response to a reorganization in the lab. At least three people reported that they modified their rules in response to a trip, because of the increased need for mail handling capabilities. In the example cited above, three external events occurred simultaneously, causing a change in his use of rules, which caused a significant change in his future mail handling behavior. Six people modified their rules shortly after an interview. (Note: By definition, everyone modified rules during the installation interviews, so these interviews were not counted.) During one follow-up interview, a person commented: *"I should redo my rules now that I've done it for a month."* Although he had thought about the improvements he wanted to make, it was not until after the interview that he actually made the changes.

Follow-up interviews often affected a user's perception of Lens, and thus their subsequent behavior. Not only did users learn about new features, but they were forced sometimes to question their use of Lens. For example, I asked people whether their rules had moved their messages correctly. Several people looked at me in surprise and consternation. It had never occurred to them that the rules might *not* work correctly and they began to worry about this new possibility.

Additions to the software, software problems that caused people to lose rules, and the conversion from one version to another also provided opportunities for people to re-evaluate their rules. Five people added the "boring" rule when it became available. Three people recreated rules differently after their rulesets were unintentionally deleted. Most people (87%) took advantage of the addition of the folder tree to "clean up" their folders, sometimes adding new ones that made sense in the hierarchy and sometimes deleting ones that were no longer deemed useful. All ten people who converted from version two to three changed their total number of rules and nine of ten also changed their rule strategy. (The person who did not change simply recreated four of the six rules he had used before.) The effect of their conversions from version two to version three of Lens is shown in Table 2.

The conversion from version two to version three shows an abrupt change in most user's trend of increasing or maintaining the number of rules. Within phase two, four users increased the number of rules and five people tried it briefly and kept the same number throughout. Only one person decreased the number of rules. In phase three, eight people increased their number of rules, one person kept the number the same and one person reduced the number of rules. However, when these users shifted from version two to version three, eight people started over with a decreased number of rules. Only one person added new rules at this point.

One possible explanation for this abrupt change is that most users do not check to see whether their rules work and thus tend not to delete ineffective rules. However, when forced to recreate their rules, they also re-evaluate them. If they are deemed worthwhile, they are created again. If not, the rules are omitted. Thus the conversion from one version to another can serve an important role in improving the overall effectiveness of a user's rules. A detailed analysis of the patterns of rule changes is presented in Mackay [1990a].

3.2 Lens users adapt Lens for their own purposes

As the previous section has shown, Lens users were influenced by a variety of factors that contributed to their interpretation and use of the software. However, they were not simply passive participants. Most Lens users were proactive as well. Although a number of users simply followed the scenarios presented to them and used Lens as it had been described, others made creative use of different features in the system, devising new uses for Lens. A few created new models of using Lens that were adopted by their colleagues. A few others actually took advantage of Lens features to provide additional functionality and were later incorporated as standard features in later versions of the software. Although Lens was not used on a large enough scale to have a major impact on the organization, one can easily see how some of these user-generated patterns of using software could influence the overall organization.

This section describes how Lens users adapted the software for their own purposes and influenced:

1. **Future use of Lens**: The creation of rules and other innovations to

Table 2: Patterns of rule changes from Version two to three

User	Version	# Rules at Start	# Rules at End	Lowest # - Highest #	Pattern of Rule Changes
A	Two	3	14	3-14	Steady increase in rules. Changed when researcher visited or on long weekends.
	Three	1	5	5-5	Follow-up interview: recreated 1. Added new rule every 2-3 days for 1 month. Added 1 after 3 months.
B	Two	14	13	12-19	Created 14. Added/deleted every 2-4 days for 10 days.
	Three	1	10	1-10	Recreated 1 rule. Added 4 more. Added every 1-2 weeks. Stable for 3 months.
C	Two	94	94	94-94	Copied rules from ***H***. Tried briefly, then stopped.
	Three	2	15	2-15	Follow-up interview: recreated 2. 3 days later, added 3. 2 months later, added 6. Added 3, at 1-month intervals.
D	Two	7	17	7-17	Highly variable, changed every 7-10 days.
	Three	17	9	2-17	Restarted with 17. 6 weeks later, restarted again with 2. Steady increase with changes every 5-15 days.
E	Two	7	7	7-7	Created 7 rules, then stopped.
	Three	12	14	12-14	Recreated 12 rules. Added 1 after 2 weeks and 1 more after 2 weeks. Stable at 14 for 8 months.
F	Two	9	11	9-11	Created 9, added 2 days later. Edited 3 weeks later.
	Three	25	39	17-39	Restarted with 25, deleted 8, added 1. Added rules once a month.
G	Two	75	98	75-98	Created 98 rules over several days. Then stopped.
	Three	23	52	23-52	Recreated 23 new rules. Added rules steadily over the next 10 months, occasionally deleting rules.
H	Two	7	7	7-7	Created 7 rules for sorting mail after reading.
	Three	1	3	1-3	Recreated 1 rule. Added 1 after 6 weeks and 1 more 6 weeks later. Stable for 6 months.
I	Two	6	6	5-6	Created 6 rules. Added 1, then deleted it.
	Three	4	7	4-7	Recreated 4. 1 added 4 months later. 2 added 2 months later. Stable at 7 for 2 months.
J	Two	23	23	23-23	Created 23 rules. Used for 2 months.
	Three	4	4	4-4	Recreated 4 rules. Accidently deleted 2 months later. Used autogetmail. Added 4 rules 2 months later.

modify their future behavior.

2. **Other users**: The sharing of innovations to help others.
3. **Lens software**: The changed models of use that influenced new versions of the software.

Mackay (1988b) reported that members of this organization use the existing electronic mail system in diverse ways. In addition to the normal functions of a message system, they use mail for personal time management, for delegation and tracking of tasks within groups and for managing and sharing large quantities of time-sensitive technical information. Not surprisingly, these same users applied the Lens technology in diverse ways to support these different uses of electronic mail. A few were very innovative and created new patterns of managing their mail.

3.2.1 Users influence their own future use of Lens

One example of a creative use of a Lens feature was by a manager who wanted to use Lens rules to help her communicate with her secretary. She objected to using formally-defined message types, because they were slow, none of the existing types were appropriate for her purposes, and she could not create her own. So, she used a customization feature in a slightly different way to accomplish the same objective. The customization feature allowed users to add commonly-used alternatives to a menu, which became available during message composition. For example, the user might specify the usernames of the people he or she commonly sent messages to. The manager wanted to identify five categories (or "buckets"), which her secretary could then act on. Rather than creating five different message types, she added five code words to the menu for a single message type: *Please file*, *Remind me*, *Action*, *Please reply to*, and *fyi*. She would then read her mail, identify a message that she wanted to forward to her secretary for action, quickly append one of these code words to a field in the forwarded message, and send it. The secretary could then use her Lens rules to identify these code words and act accordingly. This is an example of how users adjusted the system to better support communication between two people who work together closely. It also demonstrates the user's desire to annotate incoming messages. The manager could not have created Lens rules that performed this categorization automatically, because the assignment was based on her knowledge and information in addition to the specific content of the message. Thus, the manager engaged in a two-step process of manual annotation followed by automatic processing.[4]

Sometimes the effect of Lens changes as the user changes over time. For example, I interviewed a programmer who had recently joined the research lab (although not the company). At the preliminary interview, he was actively trying to identify which sources of technical information were important for his job. He had subscribed to 46 distribution lists, described himself as overwhelmed with mail and wanted Lens to help him deal with the problem. Six months later, he had stopped using Lens. He had decided that many of the distribution lists were "junk": *"I got off a lot of d.l.'s...I'm down to a few general lists."* This significantly reduced his volume of mail and he now felt in control of both of his job and the information he needed to perform it. This user's need for Lens were influenced by his changing relationship to his job.

---

[4]Note: The manager later reported that this strategy was not as successful as she had anticipated: *"I use the customized pop-up alternatives less than I expected -- the buckets are too big."*

Lens initially helped him, by letting him identify unimportant distribution lists. He reacted by removing himself from the lists altogether, which eliminated the need for Lens. Thus, the technology can have a complex, and changing, set of effects on users.

3.2.2 Users influence other Lens users

Sometimes, a single person can affect the norms of behavior a large number of people. For example, the person who discovered the ability to run rules *after* reading mail affected the mail handling strategies of 17 people and increased the number of participants in the Lens study, including several who had previously refused to try it. Two thirds of the users (12/18) of version three developed a style that incorporated both strategies of using mail.

Sometimes, one person will find an innovative solution to handling inappropriate behavior by others in the organzation. For example, one person realized that he could identify sequences of replies to a "flame". The mail system identifies replies to a previous message by placing "RE:" in the subject field and appending the rest of the subject of the original message. For example, if the header on the original message was:

To: *distribution list*
Subject: Tyrolian Ski Bindings

replies to the message would look like:

To: *distribution list*
Subject: RE: Tyrolian Ski Bindings

He decided that if the original message was not very interesting, the replies were not likely to be interesting either. So, his goal was to identify messages of the first type and delete all subsequent replies. He could have brought up a rule editor and created a unique rule for this purpose. However, he decided that it would be more convenient if he could simply press a button, as he was reading his mail, and append this new subject to an existing rule that deleted other boring messages. So, for example, if he already had a rule that deleted messages with the subject: "RE: Lehigh Safety shoes", when he pressed the button, "RE: Tyrolian Ski Bindings" would automatically be added to the list, as shown below:

If Subject: "RE: Lehigh Safety shoes" or "RE: Tyrolian Ski Bindings"
Then delete

Because he was a programmer, he was able to actually create this button and thus created the first "automatic" rule. He then influenced others in the organization; five people adopted this rule and most found it very effective: *"I use the boring rule to filter out flamers."* The developer of version three added it as a regular feature of the software. Of course, these kinds of rules need not delete the message. One user suggested: *"Why not have an 'exciting' rule, where you see an interesting message and* *automatically send it to the anyone server."* Several others wanted to use the same idea to keep track of interesting or important discussions, saving them in a folder rather than deleting them. (This was later implemented by a student at MIT.) This is a nice example of a user innovation based on a reaction to violations of the norms of behavior in the organization. It also illustrates the importance of allowing users to evaluate and annotate messages as they read them, to facilitate future processing of that message. This creates a partnership between the user and the technology, allowing users to add information that is inaccessible to the computer.

Another way users influence others is based on the roles they play in the organization. For example, mail users have two roles: senders and receivers of mail. Virtually everyone receives more mail than they send. Thus, from the perspective of the organization, it make sense to encourage senders to provide extra information, e.g. use message types, to make the recipient's mail processing task more efficient. However, senders rarely have incentives to do this. An exception is the bug report message type, which fills in information for the sender and thus saves time.

A few people send out most of the "institutional" messages which are broadcast to everyone. Thus, it makes sense to ask these people to format their messages in a standardized way. Because they repeat the task over and over, this not only saves them time and but also provides a more predictable message that is easier for the recipient to process. Good examples are the use of standardized phone messages by the receptionist and standardized seminar announcements by the secretary responsible for the seminar. Because most people in the organization *don't* send out these messages, it doesn't make sense to provide them in the sender's message type "tree". (However, it does make sense to make them available when people create rules for incoming messages.) People did not want to choose message types from a large number of infrequently-used alternatives: *"I want to suppress most message types [and use] message types that are my own."* Even phone messages, which may be sent occasionally by people other than receptionists, are unlikely to be created by ordinary users. However, several people were interested in creating message types specifically to help communication between pairs or among small groups of people. Two people wanted to create message types for their working groups: *"I want to create my own message types for my group."* and *"It would be potentially useful for [group] messages."* Three of the managers expressed an interest in using them for communication with their secretaries: *"I want to be able to send [secretary] a 'to do' list with priorities, using code words."* The key here is to make sure that the costs for adding extra information are low and the incentives sufficiently high that individuals use the technology in ways that benefit the organization. If individuals cooperate, then they can increase the overall productivity of the organization and affect the expected norms of behavior.

Rule sharing took another form as well. Two manager-secretary teams agreed on a standard form of communication and created rules to to facilitate processing messages to each other. In another case, members of a group agreed to customize their Lens rules to make their interactions with each other more efficient. These examples have a different character than the previous ones. They involve a social commitment to work together in a particular way rather than sharing primarily *individual* patterns of working with each other.

Although the number of users was quite small, and there were no facilities for exchanging rules, some users actively shared rules with each other. In each case, the rules were copied, used, evaluated and sometimes discarded. Most of the people who copied the "Lens wizard"'s rules were programmers with a range of 6 to 62 of their own rules who adopted one or two rules at a time. However, people with similar jobs and interests also borrowed rules from each other. New users found it easier to try Lens because they could try rules that had already proven useful to others. Most people benefited from this sharing, taking advantage of each other's knowledge and ideas, learning new methods of accomplishing useful tasks, avoiding errors, and generally saving time. The net result is that users adopt patterns of use from each other, propagating both useful innovations (and possibly errors) throughout the organization.

3.2.3 Users influence their Lens software developers

The original design of Lens was influenced by and derived from a set of needs at the original site. The model of use was presented as the relationship between a secretary and a manager: the secretary's (and Lens's) job was to read and prioritize messages first, and then hand the prioritized set to the manager (or Lens user). Some users chose to only use Lens *after* they had read their mail. This activity was initially enabled through a different use of a debugging feature, but subsequent versions of Lens allowed users to accomplish the same function by creating multiple rulesets triggered by different events.

The use of multiple rulesets not only changed people's behavior, it also influenced the software design. Because rules could now call other rulesets, it became important to identify the conditions under which a ruleset should stop processing a message. So, the developer of version three introduced the "stop" action. In order to avoid creating additional rules, he also allowed for multiple actions in the same rule. Thus a rule could now move a message to one or more folders, forward it, file it and mark it as "replied to". The implementation made certain assumptions about how people work; some made things easier for users and some made them harder. For example, the differences between the use of the NEXT key in Lens and Lafite caused problems for users. One of the easiest ways to see the changes in the technology is to look at the changes in the rule editor. The left side of Figure 2 shows an example of a rule created with the first rule editor. The rule has been given the name "Information Lens" and looks for messages that contain the word "Lens" in the subject field. When a message matches, Lens moves it to the folder called "Lens".

The right side of Figure 2 shows how the rule editor looked in phase three. The user interface was designed to meet the local user interface standards. Notice the addition of menu buttons and of multiple rule actions. In this example, the user has specified that messages addressed to or copied to the **Want Ads** distribution list, which also contain the subject **Honda**, are to be moved to the **Car** mail folder. Whenever a message meets these criteria, the rule fires and the message is moved to the **Car** folder. The field labeled TO OR CC: was a feature added in response to many requests by users. In previous versions, this example would have required two separate rules: one addressed *to* Want Ads and one *copied* to Want Ads.

The software developers in this study were closer to their users than most. Changes and ideas from users were fed back in interesting ways. One of the effects of this was that the users taught the developers new ideas about what Lens is, extending it in a number of different ways. Through observation of actual use, the developers examined these needs and developed a new, richer understanding of the software. Lens inspired two independent development efforts at MIT, Argus and Object Lens.

Figure 2: Lens Rule Editors

4 A User-generated Cycle of Innovation

The previous sections present isolated examples of the influences among users, their colleagues, the software, the organization, and external events. This section provides an integrated look at a cycle of innovation, in which all of these factors came into play.

One of the most interesting findings of this study was how a few users reinterpreted Lens and developed new usage patterns for themselves and their colleagues. Each new version of the software was enhanced to better accommodate each new model of use. The revised software was again reinterpreted by users, the innovations were spread throughout the user community and again influenced subsequent versions of the software. Figure 3 illustrates this user-generated cycle of innovation.

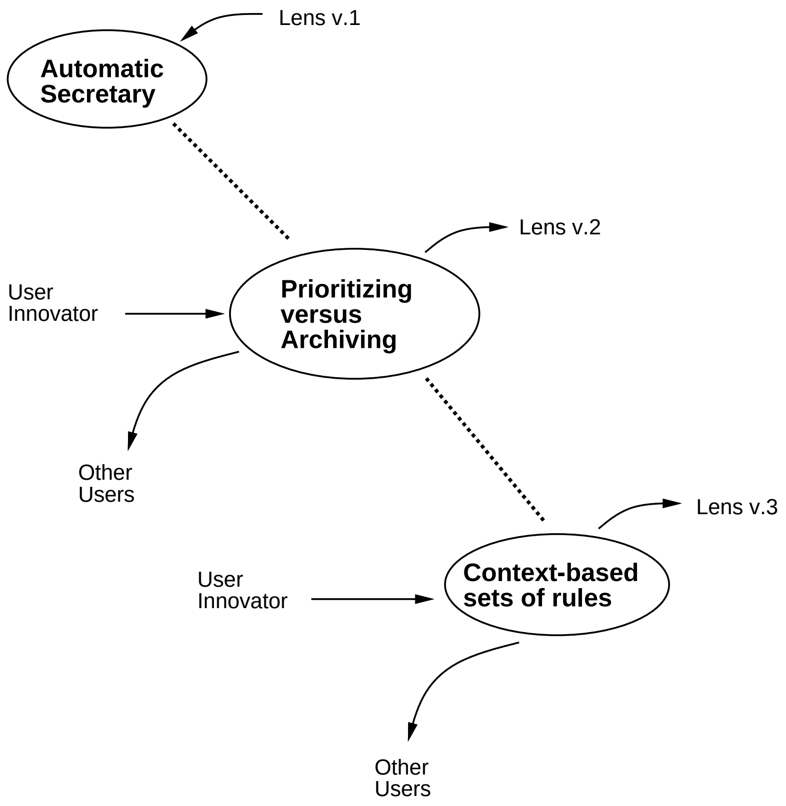

Figure 3: A user-generated cycle of innovation

The first version of Lens was described as a sort of "automatic secretary" that pre-sorts mail into prioritized groups of messages. Users could specify a set of "local" rules (a ruleset) that would be run on incoming email messages, before the user read them. Users were also given a debugging facility to test rules on a specified folder for test messages. Several potential users avoided Lens, stating they were uncomfortable with a system that processed messages before they had been read.

One user discovered he could use the debugging feature to run rules on a mail folder at any time, even after the messages had been read. He then drastically changed the style of his rules. Instead of selecting important messages to be read first, his rules sorted previously-read messages into archival folders, whenever he wanted to 'clean up' his mail. He created a new model of use, that of archiving messages.

This changed not only his perception of Lens, but also that of other users. As one person commented: *"I didn't think of reading mail and then sorting."* Many people who had initially resisted trying Lens joined the study because they were interested it this new pattern of use. Based on inspection of their rules and interviews, it was easy to determine which of two models of Lens use the users preferred: some liked the original "automatic secretary model" to prioritize incoming messages and others preferred the "automatic filing" model to archive messages that had already been read. A more detailed description of the differences between these two models of use is described in (Mackay, 1989a).

In version one, users had to choose between these two models because they were only allowed a single ruleset. To allow users to easily choose between these two models, version two was rewritten to enable users to run a ruleset at any time, before or after reading mail. To make implementation easier, the software developer also made it possible to create more than one ruleset. Once users had the option of using as many rulesets as they wished, 70% converted to using both kinds of rulesets. (One person created three kinds of rulesets). Of those who continued to use a single ruleset, one person stayed a prioritizer, one stayed an archiver, and one switched from archiving to prioritizing. This innovation thus spawned additional innovations, as users used different rulesets in different combinations to manage their mail.

A subtle aspect of multiple rulesets led to the next user innovation; the trigger for when to run the rule was now separate from the ruleset. Although automatic triggers (e.g. those that fire upon receipt of new mail or some other system-defined event) were clearly useful, users also liked the idea of manual triggers (i.e. those specified by the user). Many situations are easy for the *user* to identify (such as "I'm busy" or "I've just returned from vacation"), but difficult for the software to detect. In such situations, users found it easier to create rules without clearly articulating the conditions under which they would be run. Instead, the user would simply run the ruleset 'when the time is right'. Users were thus able to distinguish between rulesets based on self-defined changes in their personal work context, rather than characteristics available to the system. The most common example of this was the use of two sets of prioritizing rules (which several people spontaneously invented). One set was designed to handle mail on a normal day; the other would use much stricter criteria use when returning from a trip or when mail had piled up. These innovations again caused users to revise their overall view of Lens and were shared within the organization.

The use of multiple rulesets generated another insight about how users interact with mail. In the original design, rules were assumed to operate on messages based solely on content, length, or other attributes found in the message, independent of the user. However, consider what actually happens when a user receives a message. When it first arrives, the message is "new" and "unread" and potentially interesting. The level of interest may be affected by many factors, such as the user's current state (e.g. extremely busy versus bored) or the quantity of other messages (e.g. the only one versus one of a hundred). Most users scan the message header to determine whether the message should be deleted, acted upon immediately, or left for later. If the message is read and requires a reply, the length of time between the message's arrival and the current time becomes important. Thus, if the message arrived 20 minutes ago, the user is unlikely to be concerned. On the other hand, if it arrived two weeks ago, the user begins to feel guilty and feels that it is important to reply soon. Yet, this is a U-shaped curve: if the message is three months old, the reply probably doesn't matter anymore. (The guilt may fade more slowly.)

Users clearly adapted to the software base on how it was presented to them, their previous experiences with mail and other software, their on-going experiences with Lens and by the experiences of others in the organization. A few people also innovated in ways sufficient to change the mail processing strategies for others in the organization. Users took advantage of different features; a few used features in ways that were not anticipated by the original software designers and were not supported in the functional specification.

Each new version incorporated changes made by users who "broke" the previous software model and caused new changes in the patterns of use throughout the organization. Such a tight iterative design loop is unusual: the different views of the software created by users continued to be incorporated into subsequent versions of the software.

This user-generated cycle of innovation did not occur just once. The basic pattern consisted of the introduction of a new technology, presented with a particular model of use. An individual user found a way to use the software in a different way, creating a new model of use. Others in the organization adopted the revised model and the software developers made changes in the software to facilitate this new type of use. The revised software again provided an opportunity for an individual user to innovate and create still another model of use. Again, others adopted the new model and the software developers were again able to revise the software to facilitate the new model of use.

Proponents of participatory design strategies and iterative development concentrate on the initial development of the software. This study illustrates the importance of participant involvement at the other end of the process -- when the software is actually in use in the organization. End users can provide important insights into new uses of the technology, because they are in direct contact with the needs of the users. User-created innovations are based on real needs and provide practical solutions to problems that people face in the everyday world. New uses can provide an important source of tested ideas for future software.

Where does this go?

It is difficult for people to articulate appropriate behavior in sufficiently explicit terms for the computer. There are several reasons:

1. People rarely have access to data about their own behavior and their perceptions of their own behavior are often wrong.

   For example, Mackay (1988d) showed that none of the users were able to accurately estimate the quantity of distribution lists, and few were able to estimate the quantity of mail folders. Similarly, user's predictions of their future behavior, obtained during the preliminary interviews, often differed significantly from their subsequent use of Lens.

2. People have difficulty articulating their behavior and translating their desires into rules. When asked, they can make up rules, but these do not always correspond with their actual behavior. They usually refer to an "ideal" processing state and include a variety of misperceptions about technology and their own behavior.

   For example, one person commented about the *anyone* server: *"I wouldn't know what to ask for."* Another said: *"I couldn't think of anything useful...It's a conceptual problem. I couldn't think of the different sorts [of messages]."*

3. Some of the factors necessary for a particular decision are not accessible to the computer.

   For example, decisions about which messages to delete change according to how busy the user is at the time of the decision.

4. User's actions depend on the context in which the decision is being made.

   Suchman (1988) argues that the reason people find it difficult to articulate their plans of action is that "plans are resources for situated action, but do not in any strong sense determine its course...rationality anticipates action before the fact, and reconstructs it afterward" (p.52,53).

****

People spend varying amounts of time organizing their work, as opposed to doing it. Filing, creating piles, scheduling work, organizing papers, etc. do not accomplish work per se, but allow people to prioritize activities and get important things done. If the input appears faster than the ability to process, even if only in cycles, then this is necessary (Weick, 1988) Sometimes these activities serve to remind people about what they need to do next, make certain things easy to find, or simply place like projects together to reduce the cost of switching between tasks.

Another observation is that users who described themselves as "overloaded" were less likely to spend time using tools that helped them reduce overload. Not surprisingly, people who felt very much in control were also less likely to spend time organizing (they had either already invested the time and were reaping the benefits or their jobs or personalities were such that they were able to manage the flow of tasks/information.) The most interesting group were the people who were "on the edge" - basically handling things but at their limits. These people were able to try new things and were more innovative than their peers. They had a higher need than the people who felt in control, thus the benefits were higher. They were also more willing to pay the costs of organizing, because they weren't completely overloaded. Panic induces paralysis.

Users who have very little incentive to modify won’t because the benefits aren’t worth the costs. Users who are overloaded have the incentive, but are not willing to risk the possibility of losing it all. Thus, we expect people who are in a state in which they are basically in control, but nearing the edge, to be the most likely to innovate/modify their working environments. Notice that this changes over time. When a user comes in contact with a new piece of software, the tendency is to move into "survival" mode first. The goal is to replicate the set of tasks that the user was able to accomplish before and try to get the job done. Over time, as that set of tasks becomes familiar, the user can do them more easily. At this point, the user moves from an "overload" situation to an "in control" situation, and is able to pick up more. Some users will stop here, others will begin to increase the demands of the job, taking on more responsibilities, etc.

5 Discussion

The results of this study demonstrate that Lens users did not simply react to a static technology; they actively changed it. They were influenced by how the software was presented to them, by their previous experiences with the existing mail system and other software, by their on-going experiences with Lens and by the experiences of others in the organization. For a few people, Lens fundamentally changed their mail processing strategies. Different users took advantage of different features; a few used features in ways that were not anticipated by the original software designers and were not supported in the functional specification. These activities influenced both the subsequent design of the software and how other members of the organization thought about and used Lens.

Not only did the technology change, but users changed too. Instead of acting as passive recipients of the technology, they actively reinterpreted it. In some cases, they modified the software to create innovations that affected the future development of the technology. Patterns of use were propagated throughout the organization and members of groups adopted different views of the technology.

merge these two parts

This study has demonstrated some of the diverse ways in which individuals react to and appropriate an information technology, the Information Lens. Each person uses some features and disregards others; two people may use completely non-overlapping sets of features. Previous experience with other software, differences in job requirements and mail handling patterns, and differences in personal characteristics all affect the user's expectations about what Lens is and can do. Perceptions about Lens and perceptions about their own needs affect their subsequent interactions with the software and the effectiveness of those interactions affect their on-going perceptions of Lens and Lens use. By observing how users actually used Lens, the developers were able to re-examine their own perceptions of the software and enhance subsequent versions of Lens to take more of the user's needs into account.

The sharing of rules is a subset of a larger set of issues, including how users learn to use new technology and the ways they innovate and appropriate the technology for their own purposes. As users learn a new technology, they are influenced both by its design and the established patterns of use by other members of the organization. At the same time, individuals innovate. By sharing these with other members of the organization, they act to affect the structure of the organization. Software does not remain static when it is introduced into an organization. People in the organization evolve their individual patterns of use, share them with each other, react to external changes, both technical and non-technical, and sometimes proactively modify the system to produce significant innovations.

The previous results make it clear that users did not simply "learn" the technology presented to them and adopt it. Instead, a very complex set of changes took place in the individuals, the technology, and the organization.

5.1 Co-Adaptive Systems

I have introduced the term *co-adaptive phenomena* to describe the interaction between individual users and technology. "Co-adaptive" emphasizes the dual nature of the phenomenon: that people both adapt to technology (reacting to it) and adapt it to meet their needs (proactively changing it). Both processes occur over time and influence each other. The traditional scientific approach has been to isolate naturally-occurring phenomena and study them under controlled conditions. Several researchers (see Orlikowski and Baroudi (1989) for a discussion of these) have pointed out that the research methodologies designed to understand these phenomena are not appropriate for the study of technology, which is both the product of and is changed by human action.

This study provides a very concrete example of a co-adaptive phenomenon. Such phenomena pose theoretical and methodological challenges and are difficult to study in traditional ways. However, some aspects of the interaction between people and technology only make sense when such phenomena are taken into account. We can learn from changes in the research methods in some of the natural sciences. For example, evolution is a natural phenomenon in which plants and animals adapt to their environments. However, more recent research has identified a secondary phenomenon: plants and animals actively change the environment and these changes affect subsequent adaptations. The combination of these phenomena has been called *co-evolution* to stress the dual nature of evolution (Lovelock, 1979).

Scientists who study the atmosphere, e.g. at NCAR, the National Center for Atmospheric Research in Boulder, Colorado, have had to move to a multidisciplinary approach. They have concluded that in order to understand the earth's atmosphere, they must understand two related sets of phenomena: how plants and animals are affected by changes in the earth's atmosphere, and how their adaptations to those changes cause corresponding changes in the atmosphere. This research requires a global perspective and an interdisciplinary approach: "The role of the biosphere, the living material on the earth, is once again recognized as contributing to myriad chemical reactions, and new geochemical data are revitalizing research into the origins of the atmosphere...NCAR scientists, in collaboration with colleagues elsewhere, are now beginning the important work of linking our atmospheric models with equally complex and equally important biological, chemical and oceanographic models." (NCAR 1985 Annual Report, p.14, p.65)

An example of the successful use of this approach was in the discovery of the relationship between "biomass burning" in underdeveloped countries and the atmosphere:

> *The developing world resorts to clearing marginal farming land...by burning it off...This type of burning, which could be called nonindustrial pollution, may produce as much carbon monoxide, methane, and other gases as does the developed world...Crutzen led members of the NCAR Biosphere-Atmosphere Interactions Project in a tropical field program in Brazil called Project Brushfire in 1979 and 1980. Since 60% of the world's biomass resides in the tropics, this region has the potential for large emissions that can have global atmospheric effects...The measurements indicate that CO emissions from fires may make up as much as one-third of the global CO budget.*

Scientists were puzzled as to the source of high concentrations of methane in the atmosphere. They also observed the appearance of large numbers of termite mounds in the areas cleared by biomass burning.

> *During the biomass burning experiment, Zimmerman noticed that termite mounds were ubiquitous and especially common wherever fields had been cleared...Thinking that termites might prove to be a significant source of atmospheric methane...Zimmerman did field work in Guatemala and Africa to study the emissions of different species of termites. Laboratory studies at NCAR confirm that termites do release methane and can contribute significantly to the atmospheric methane budget.*

They eventually determined that the methane released from termite mounds, which has increased due to the burning of farmland, account for 24% of the earth's methane. Of course, these changes in the earth's atmosphere are having long-term effects on people and animals, who must adapt to the changes in the atmosphere or not survive.

Living organisms do not adapt to a static atmosphere, they adapt to a dynamic one and which they actively participate in changing. Their adaptations change the composition of the atmosphere, which in turn affects their subsequent adaptations. Just as we cannot think of the earth's atmosphere as a fixed variable that operates in predictable ways independently of living organisms on earth, we cannot look at technology as a fixed variable that operates in predictable ways independently of the people who use it. People adapt to new technologies, but they also appropriate and change them, which affects their future interactions with the technology. This study has shown that we can view the interaction between users and technology in an organization as a co-adaptive phenomenon.

This research has both theoretical and practical implications for the study of how people interact with technology. The theoretical questions include: What research models provide a useful way to study these phenomena? How can we better understand the ways in which people are influenced by the software they use? How can we better understand the ways in which people appropriate technology for their own purposes? The practical questions include: Which users are most likely to innovate and under what conditions? How do individual innovations affect group norms of behavior? What are the implications for the design of software and its introduction into organizations?

Is it also possible to explicitly create productively co-adaptive systems? I believe we can, by affecting software designs and the process of designing, developing and introducing software into organizations. We can create reflective software (e.g. Kiczales et al., 1991) which may provide feedback to users as well as to the objects in the system. We can develop new models of customizable software (Trigg et al., 1987, Maclean et al., 1990, Mackay, 1990b) which allow users to capture and modify patterns of interaction. We can also provide mechanisms to facilitate sharing of these patterns and other innovations (Nardi & Miller, 1990, Maclean et al., 1990, Mackay, 1990b), to help distribute innovations within an organization.

Coadaptive systems include change at many levels, both technological and human. user innovation is just one thing that can happen...and it doesn't happen particularly often. Still, when it does, the payoff may be high. How do we capture it? One view of innovation is simply that the user brings a different set of assumptions to the problem (or activity) and thus reinterprets it in an interesting way.

I have defined the class of such phenomena as *co-adaptive*, to capture the idea that people may both adapt to some aspect of the environment (e.g. computer technology) and at the same time adapt that aspect of the environment for their own purposes. Co-adaptive phenomena pose theoretical and methodological challenges and are difficult to study in traditional ways. Yet, some aspects of the interaction between people and technology only make sense when such phenomena are taken into account.

This paper provides a concrete example of the co-adaptive use of technology, based on a longitudinal study of the introduction of new software application into an organization. In particular, I am interested in how user innovations can affect the future design of the technology and its future use within the organization. Finally, the paper will discuss the implications for research in HCI -- how can we deal with the dynamic nature of HCI, in which the artifacts being studied are created by people and people change and recreate the artifacts.

The structurational model proposed by Orlikowski (1989) takes both perspectives into account and incorporates an active role by individuals in the organization. This paper extends her analysis by examining the *co-adaptive* relationship between users and user-customizable software: users both adapt to the available technology and appropriate the technology, adapting it over time. These appropriations may take the form of user innovations which may change both the technology itself and the characteristics of the organization, such as who communicates with whom and how coordination of work processes is handled.

Section xx describes how these results challenge the original conception of the research and provide support for the claim that the use of information technology is a co-adaptive phenomenon. Section x.x discusses the implications for future research.

As software technology has changed, so has research about it and its use in organizations. Because the technology is a moving target, research strategies borrowed from the natural sciences that define the problem as predicting and controlling isolated natural phenomena have not met with much success. I am interested in the problem of how to study phenomena that *cannot* be isolated in this way, because they change as they are used.

Then -- what do we make of this? First, it has implications for the way we design studies and what we look for. But secondly, it challenges the basic groundwork on which much of the research is based. If the users modify the technology, then you can't simply look at how the technology affected them. Especially since, once they've modified it, it's different technology. And it doesn't stop -- it keeps changing and getting reinterpreted. Sometimes in big jumps, sometimes in small ones.

References

Allen, T.J. (1972). Communication Networks in R&D Laboratories. *R&D Management*, , pp. 14-21.

Barley, S. R. (1986). Technology as an Occasion for Structuring Evidence from Observations of CT Scanners and the Social Order of Radiology Departments. *Administrative Science Quarterly*, *31*, 77-108.

Brown, J. and Van Lehn, K. (1980). Repair Theory: A Generative Theory of Bugs in Procedural Skills. *Cognitive Science*, *4*, 379-426.

Card, S. and Henderson, A. (April 1987). A multiple, virtual-workspace interface to support user task switching. *CHI + GI Conference Proceedings*. Toronto, Canada.

Chapanis, A. (1969). *Research Techniques in Human Engineering*. Baltimore, Maryland: John Hopkins Press.

Dick and Carey. (198). *Principals of Instructional Design*. : .

Ehn P. (1989). *Work-oriented design of computer artifacts*. Hillsdale, New Jersey: Lawrence Ehrlbaum, Assoc.

Eveland, J. and Bikson, T. (September 1988). Work Group Structures and Computer Support: A Field Experiment. *Proceedings on the Conference for Computer-Supported Cooperative Work*. Portland, Oregon.

Giddens, A. (1984). *The Constitution of Society: Outline of the Theory of Structure.*. Berkeley, California: University of California Press.

Gould, J. & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(7), 300-311.

Greenbaum, J. and Kyng, M. (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, N.J.: Lawrence Ehrlbaum Associates.

Kling, R. (1980). Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research. *Computing Surveys*, , pp. 61-110.

Kling, R. and Iacono, S. (1984). Computing as an Occasion for Social Control. *Journal of Social Issues*, , pp. 77-96.

Lai, K. and Malone, T. (September 1988). Object Lens: A Spreadsheet for Cooperative Work. *Conference on Computer-Supported Cooperative Work*. Portland, Oregon: ACM.

Lovelock, J.E. (1979). *Gaia: A New Look at Life on Earth*. Oxford, England: Oxford University Press.

Mackay, W.E. (August 1986). Lens Reference Manual and User's Guide. Massachusetts Institute of Technology.

Mackay, W.E. (October 1988). Diversity in the Use of Electronic Mail: A Preliminary Inquiry. *ACM Transactions on Office Information Systems*, Vol. *6*(4).

Mackay, W.E. (May 1989). Tools for Supporting Cooperative Work Near and Far: Highlights from the CSCW Conference. *CHI '89 Conference on Human-Computer Interaction*. Austin, Texas: ACM/SIGCHI, Panel presentation.

Mackay, W.E., Gardner, B.R., Mintz, T.H., Pito, R.A., and Siegal, J.B. (August 1989). Argus Design Specification. Distributed by the Open Software Foundation and MIT Office of Technology Licensing.

Mackay, W.E. (May 1990a). *Users and Customizable Software: A Co-Adaptive Phenomenon*. Doctoral dissertation, Massachusetts Instititute of Technology.

Mackay, W.E., Malone, T.W., Crowston, K. Rao, R., Rosenblitt, D., and Card, S. (May 1989). How do experienced information lens users use rules? *CHI '89 Conference on Human-Computer Interaction*. Austin, Texas: ACM/SIGCHI.

Malone, T.W., Grant, K.R., Lai, K.Y., Rao, R. & Rosenblitt, D.R. (1987). Semistructured Messages Are Surprisingly Useful for Computer-Supported Coordination. *ACM Transactions on Office Information Systems*, *5*(2), 115-131.

Malone, T.W., Grant, K.R., Turbak, R.A., Brobst, S.A., & Cohen, M.D. (1987). Intelligent Information-Sharing Systems. *Communications of the ACM*, *30*, 484-497.

Markus, L. and Robey, D. (1988). Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science*, *34*(5), 583-598.

NCAR. (1985). *National Center for Atmospheric Research Annual Report* (Technical Report). Boulder, Colorado: University Corporation for Atmospheric Research.

Norman, D.A. and Draper, S.W. (1986). *User-Centered System Design*. Hillsdale, New Jersey: Ehrlbaum Associates.

Orlikowski, W. (1989). The Duality of Technology: Rethinking the concept of technology in organizations. Unpublished manuscript.

Orlikowski, W. and Baroudi, J.J. (1989). *IS Research Paradigms: Method Versus Substance* (Technical Report). Cambridge, Massachusetts: Sloan School of Management, MIT. Sloan Working Paper # 3028-89-MS.

Sproull, L. & Kiesler, S. (1986). Reducing Social Context Cues: Electronic Mail in Organizational Communication. *Management Science*, *32*(11), 1492-1512.

Suchman, L. (1987). *Plans and Situated Actions*. Cambridge, England: Cambridge University Press.

Weick, K. (198?). The Twigging of Overload. In (Ed.), . : .

Zuboff, S. (1988). *In the Age of the Smart Machine*. New York: Basic Books.

Table of Contents

List of Figures

List of Tables