Information Needs in Technical Work Settings and their Implications for the Design of Computer Tools

ANDREAS PAEPCKE

Stanford University, Gates Computer Science Rm. 426, Stanford, CA 94305
paepcke@cs.stanford.edu

Abstract

We interviewed information workers in multiple technical areas of a large, diverse company, and we describe some of the unsatisfied information needs we observed during our study. Two clusters of issues are described. The first covers how loosely coupled work groups use and share information. We show the need to structure information for multiple, partly unanticipated uses. We show how the construction of information compounds helps users accomplish some of this restructuring, and we explain how structuring flexibility is also required because of temperamental differences among users. The second cluster of issues revolves around collections of tightly coupled work groups. We show that information shared within such groups differs from information shared across group boundaries. We present the barriers to sharing which we saw operating both within groups and outside, and we explain the function of resource and contact broker which evolved in the settings we examined. For each of these issues we propose implications for information tool design.

Key words. Digital Library, Groupware, Information sharing, Information compounds, Collaborative work, Information brokers, Knowledge workers, Information search and retrieval.

1. Introduction

"Computers are terse and unfriendly; humans are scattered and have ego problems." This is how a technical support engineer described to us the choice between using computers or human beings as sources of information. We wanted to find out more about the information habits of workers like him, and therefore used a mixture of interviews and observation to survey a broad range of information-intensive technical work settings. The goal was to identify some of the information-related problems that were common to workers in these contexts, and then to derive lessons for designers of information production, dissemination, display and maintenance tools. We therefore intentionally chose participants from very different work domains to find out whether guidelines for generic group information tool designs could be distilled from their practice. For this article we have selected from our data the issues we found most commonly relevant across work domain boundaries.

For example, we found that participants in the different domains needed to build what we call *information compounds*. These are pieces of information extracted from multiple, possibly heterogeneous sources and combined into a new composite structure. These compounds turned into new information artifacts that took on a life of their own. They were sometimes used for purposes other than what the constituting information was collected for. We explore some of the subtleties tool designers would need to consider in supporting tools for information compounds.

Beyond the purely technical implications, we paid attention to design requirements growing from personal preferences of the persons we interviewed. For example, we found that there were differences in temperament that had implications for design tools. There were 'detectives' who were quick, intuitive, and very impatient. In contrast, we found 'librarians' who were much more methodical, tolerant of organizational requirements, and excellent producers of useful and sharable information. We point to some requirements needed to make tools useful for both of these populations.

Reflecting our data, the article is organized into two main parts. The first explores issues that come up in loosely coupled work arrangements as found in large companies. In these settings workers do share a global goal, such as ensuring profit through effective design and manufacture of a particular portfolio of products. But participants are re-assigned to different specific tasks over time. Relationships across the more tightly coupled subteams change with the shifts in personnel. Personal involvement among participants in loosely coupled environments is limited by physical distance or the small overlap in purpose and interest.

In contrast, the second part of the article is concerned with information-related issues arising in teams that are more tightly coupled. These might be a small group of mechanical designers working on a chassis for computer printers, or they might be a group of customer support engineers all helping customers with a particular set of a company's software products. These teams work together on a daily basis, often in the same physical environment.

From the large amount of data, we selected three issues as most prevalent in the loosely coupled settings. The first revolves around the fact that information is often re-used for multiple, often unforeseen purposes. Section 2.1 argues that tools must therefore be able to recast information into different structures to accommodate multiple purposes. Section 2.2 illustrates how our informants struggled to achieve multi-purpose information reuse by building custom information compounds with little tool support. Section 2.3, finally, goes on to show that information must be restructurable not just for different purposes, but also for different types of human consumers.

For the second part concerned with tightly coupled work arrangements, we explain in Section 3.1 how the kind of information shared within tightly coupled settings is richer and often more difficult to support with tools than information shared across loosely coupled groups. Section 3.2 goes on to point out barriers that exist even within tightly coupled work arrangements. Section 3.3 then goes on to explain how the role of human 'contact broker' alleviated some of the problems of inter-group information barriers.

1.1 Methodology

The focus of our study was on identifying and analyzing information-related problems and issues that were common to different work domains. Traditional ethnographic studies rely on researchers' immersion in the setting under study over a long period of time, which would have been difficult in a multi-domain study. Interview-based approaches have also been shown to elicit valuable insights for researchers and designers (e.g. (Wixon, Holtzblatt, and Knox, 1990; Holtzblatt and Jones, 1990; Nardi and Miller, 1990; Nardi et al., 1993)). We believe that an extended discussion in the worker's office is the best approach to interviewing, since it allows the researcher and informant to see and interact with the artifacts in the work setting. The physical surroundings (including computers, co-workers, paper notes tacked up on office walls, and the like) effectively frame the topics under discussion and help to reveal and illustrate the relationships among them.

The interviews were semi-structured. By this we mean that we had a set of written questions which we prepared in advance and covered in each interview, but we did not read the questions aloud or otherwise present them directly to the informant. Instead, these questions were used as guides during the conversation. Often, conversations took unexpected turns which we were quite willing to follow, but we always made sure we covered the basic material in our prepared question list. These questions focused on job activities and both successful and unsuccessful information usage from the informant's point of view. In one setting (technical support), we were also able to observe some of the support engineers handling customer calls. During most of our interviews, we were given guided tours of the physical setting, on-line systems, paper filing systems, and other information sources.

We chose informants who had a certain knowledge worker profile. In particular, we looked for employees with jobs that required the active use and production of complex information. Within each work group, we chose people who seemed to have complementary roles or characteristics.

We interviewed thirteen people in six work groups in a large computer manufacturing company, and fifteen people in one large work area in the same company. Most of the people we interviewed performed technical jobs, though a few did not. Only four of our informants were managers. Though managers are certainly knowledge workers, we chose in this study to concentrate on technical contributors.

Informants were engaged in the following types of work: technical support, marketing, software integration, finance, electronics product design, the design and manufacturing of computer printers, and software tool design for chemical analysis lab workers. In this last case, we were interested in the information needs of both the tool designers and their customers, the lab workers. Below, we describe our informants in each of these work areas in more detail.

In the **technical support** area, one group served a large community of mostly external commercial customers and one served a much smaller community of internal, research-oriented customers. The external support group consisted of engineers located in different countries who were knowledgeable about a wide array of products. They consulted with their customers over the telephone. Their on-line data sources were combined in a single knowledge database which was accessible to all support engineers. This database was constantly updated with call histories, known bugs, product newsletters, and other relevant material. In addition, the support engineers consulted product documentation and other people for help. All engineers were equipped with workstations.

The **internal support** group was more of a “mobile force.” They were collocated with their customers and often responded to phone calls or electronic mail requests by walking to the problem site or logging into troubled machines. They knew their customers, at least by sight, so contacts and information exchange were informal. Their on-line sources of information consisted of electronic bulletin boards, diagnostic or monitoring programs, and product documentation. Customer requests and solutions were often handled through electronic mail.

The **marketing** group was engaged in three major activities: analysis of competitors' products, participation in product planning, and presentation of company products to the outside world. Typical information-related tasks were the storage and dissemination of competitive product information and the collection and packaging of information on new company products. Information sources were technical literature, electronic bulletin boards, the local rumor mill, and various company databases. The group also used the telephone and electronic mail to exchange information with coworkers.

The **software integration** group was responsible for coordinating releases of software systems whose components were produced in various parts of the company. They received information through file transfers, electronic mail, and the telephone. Most information was stored in one database which was set up and used by the integration group and was also accessible to software contributors. Information not in the database included executable scripts and text-based descriptions of the integration process.

The **finance** group kept track of purchases from outside vendors, ensuring timely payment of bills. They used their own database as a major source of information, in addition to the flow of paper from the vendors.

The **electronics product design** group was engaged in the design and manufacture of measurement equipment. It was not the only group engaged in this activity, and there was interaction with other groups in different parts of the country to exchange product and design information. Their sources of information included product definition documents, design notebooks, CAD models, and a company-wide electronic parts database.

The **chemical analysis lab** was concerned with monitoring the production of chemical compounds to ensure compliance with federal regulations. Their equipment consisted of personal computers connected to chemical analysis instruments which produced large amounts of data. In addition, large volumes of paper-based federal regulatory material were consulted regularly.

In the final setting, multiple groups cooperated on both the **design and manufacture of computer printers**. Outside vendors provided specialized services, sometimes in close collaboration with internal design groups. Information sources here were very similar to those in the electronics products group.

1.2 Analysis

To analyze the interviews, we systematically looked through the data for scenarios of information use or production. We then examined these scenarios for common threads or themes, with special attention to the different characteristics, problems, uses, and constraints of information handling. At the end of this process, we had a list of roughly twenty themes that had emerged from the scenarios, each of which was potentially relevant to designers of information systems or applications. These themes ranged in scope and topic, but all of them revolved around information needs which were at least partially unmet. Three examples are the need for moving between paper-based and on-line information sources, mobile access to information, and flexible browsing and querying interfaces.

We filtered these themes to find those which appeared in multiple settings, since our interest was in identifying common problems or issues related to basic information infrastructure. We did not prioritize the problems we saw or ask our informants to rank problems. It is important to note that only those information-related issues that were actually raised during the interviews or observations are addressed in this article. Clearly there are other important information problems, such as security and privacy, that are critical to the designers of information systems. Their absence here does not indicate a belief on our part that they are unimportant, just that they did not emerge from the many scenarios our informants described to us.

2. Loosely Coupled Cooperative Work Arrangements

We learned about the following event in the printer product development setting. It illustrates the difficulties that arise typically in large companies with an overall common goal, but with work arrangements that are large scale, loosely coupled and heterogeneous in organization, work processes and, frequently, in priorities.

A new printer was to be designed that was to be less costly than the current model. Analysis had shown that the paper feed was one of the most expensive parts of the current model, so an engineer was assigned to redesign it. His first problem was to understand why paper feeds were currently designed the way they were. Acquiring the necessary understanding took him four months. Much of that time was spent identifying and collecting relevant information, including old project documentation and the lab notebooks of engineers who had worked on previous projects. The particular work groups no longer existed. Participants had left or been reassigned. He culled from this stack of material everything related to paper feed design. Most of these documents had been created for reasons other than what the engineer needed to use them for, making their structure less than optimally suited.

This engineer's problem stemmed from the fact that while his product division overall was cohesive and moved towards a common goal—profit from well-designed printers—the structure holding the participating workers together was shifting over time. The coupling among its elements was loose when viewed over the lifetime of sequences of products, rather than a single product cycle.

In this section we touch on three of the aspects that grow from the realities of loosely coupled cooperative settings. First, we highlight the challenge for tools to support the customized structuring of information for multiple uses. Then we show how we saw *information compounds* used for coping with loosely coupled settings. Finally, we present the problem tool builders face with individual temperamental differences among users.

2.1 Structuring Information for Multiple Uses

When information providers can predict all uses of information, an optimal data model and organization can in theory be chosen for storing the data to ensure that querying, filtering, and browsing are straightforward and well matched to the users' tasks. Unfortunately, it is seldom the case that all the uses of information can be anticipated. The difficulty of structuring information for multiple uses contributes to the gap between information systems and users working on their tasks.

Traditionally, major structural decisions have been made by database designers through their choice of data model. Relational databases, for instance, predetermine much of the structure of the information they will contain by providing a table model. The structuring tasks that remain within that model are mostly reserved for information providers. Providers determine the relations and their attributes if relational databases are used; they design the class hierarchies in object-oriented databases. Information consumers have very little control over structure. The only mechanism typically available to them is the construction of views that stay within the constraints of the data model.

But different tasks require different structures for the data. For example, competitive analysis data produced by our marketing setting was shared both with research and development groups and with upper management in different formats and to different ends.

> We spend a lot of time trying to figure out ways to represent [the information] so that management can understand it. Okay, that's the most difficult thing is, say you have this list of features... but management wants to see a quick and dirty diagram that shows them: "Oh, it's very obvious to me that we need to implement x, y and z and we'll be fine."

The R&D group wanted to understand technical details of the data, while management wanted the big picture. Questionnaire data developed by marketing to evaluate customers as potential beta sites was later reused to help with customer needs analysis, by sifting through it to find relevant customer comments. A single schematic model in a computer-aided design system was used by different engineers (or the same engineer at different times) to evaluate 'assemblability', space constraints, heat constraints, layout, and necessary parts or subassemblies; each of these evaluation tasks required a different view of the information encapsulated by the design model. Even when information was reused by the same individual, structuring and organizing it was difficult for this technical support engineer:

> The basic problem I have in trying to organize my information is how to organize the information ahead of time, how to categorize it, and any category system that I devise is insufficient or misses something later on.

The problems arising when information is produced and maintained by multiple actors and then used by parties with differing interests are observable in other domains as well. In the public health domain, for example, the International Classification of Diseases has long been subject to the divergent interest of its users. (Bowker and Star, 1991) describe how users such as researchers, the police, insurance agencies or pharmaceutical companies want different levels of granularity, and often different categorizations.

The infrastructure of the World-Wide Web (Campione and Walrath, 1996) has spawned renewed attempts to address the problem of organizing information to support multiple uses. Individuals and organizations frequently create documents consisting mostly of links that simply pull together thematically-related material already available, and often created for varying purposes. Going beyond this 'hotlist' baseline, annotations and guided tours are beginning to make their way into the system (Röscheisen, Mogensen, and Winograd, 1995). In some of these annotation facilities, users can construct multiple, distinct, consistent sets of annotations on existing documents, without having to modify the document files themselves. Guided tours allow browsers to add buttons to existing Web pages, again without modifying the original document files. These buttons point the user to related material the original author was unaware of or, more importantly in this context, did not believe to be relevant for the anticipated use of the document. Links were therefore not included in the original.

For domains where information is likely to be used in multiple, unanticipated ways these observations can suggest a strategy for developers of information systems: developers need to emphasize support for users building their own organizations and structural metaphors on top of the physical storage provided by the system. Users could then construct new structures for the information as new uses arose. One way to do this is through the creation of information compounds for different users, roles, and tasks. Compounds might be specified directly by their users, or their creation could be the responsibility of information access experts in consultation with users.

2.2 Custom Information Compounds

One activity we observed frequently was the attempt to unite related pieces of information into one collection that either helped solve a particular problem or could be used as an end product for a task. One example were 'how-to' instructions for computer interactions. Those were collected from manual pages, email messages, personal notes or oral communications. They were then either used by the creator, or they were created by an expert and disseminated among colleagues.

We call these collections *information compounds*[1]. The information compounds we saw were often used repeatedly, though sometimes they were abandoned later because they had had only temporary value. In all the systems we saw, compound-building was inadequately supported. For example, it was possible to use document processors to build free-form compounds. But in most settings, there would have been no quick way of indexing the material to make it searchable and sharable. Alternatively, in one of the settings, compounds could have been entered into a shared, searchable text database. But there was no adequate support for cross referencing.

The one-time, permanent collation of material for quick reference is one use of information compounds. Fully developed information compounds would help users specify task– and role-based selections of information from large data stores. Different selections might, for example, be bound to particular compound components, depending on whether the viewer is a manager seeking an overview, or a engineer looking for details. This would support the restructuring of a shared data pool for different uses.

The components of compounds in a fully developed system would have the potential of being active. They would transparently update themselves from the original sources, when appropriate. Compounds would thus provide a filtering function by limiting the information that was retrieved, and by presenting it in a meaningful way.

For example, one support engineer estimated that at any given time, about 80% of his customer problems were covered by about twenty different queries. Even though over time the precise twenty queries changed as products moved through their life cycles, a tool would be made more valuable if it provided facilities to record and reuse these twenty queries. Such a pattern represents tangible retrieval expertise. A possible implication of this observation is that as soon as such sets of patterns are perceived, users should be able to have the system save them as part of active information compounds so that they can be reused or shared with others.

The particular engineer in the above example had compensated for the system's lack of support for capturing retrieval expertise by creating a binder into which he put information that answered the most frequently asked customer questions. This information was a collection of compounds, made up of product manual excerpts, mail messages, hand-written notes, transcripts of login sessions, and useful bits of information from the knowledge base. This was a slowly evolving source, not directly linked to the on-line systems the engineer used to do much of his work, but nevertheless sufficiently valuable that he took the time to create and maintain it. An example of an early system that tried to support some compound building as part of data source browsing activities is described in (Motro, D'Atri, and Tarantino,1988).

Note that compounds of information from different sources should not necessarily hide their inherent heterogeneity by making their components appear to have come from a single repository. This degree of source integration has drawbacks. Source transparency would destroy important information: users in several groups wanted to know where the various pieces of information they used were coming from.

For example, in the technical support group, some engineers judged the relative strengths and weaknesses of overlapping records of previous problems by looking at who had created the records, when they had been created, the policies used to verify entry accuracy in that particular source, and the format used to describe the information. The engineers were able to use that knowledge to improve the quality of information they extracted. They chose to search a limited set of data sources (such as previous customer calls, certified entries, or known bugs), or they filtered entries based on attributes such as the author's name. If the engineers had had the ability to assemble on-line information compounds of the materials they used, they would certainly have wanted the provenance of each component to be apparent. Analogous findings are reported in (Cicourel, 1990) where an analysis of decision making processes among physicians showed evidence that the sources of information and the physicians' assessment of source quality were important factors.

In the remainder of this section, we describe the support needed for building compounds and using them to capture retrieval expertise and create multiple logical structures for information. These discussions are based on the manual compound-related activities we saw across domains.

2.2.1 Building Compounds

Information compounds are difficult to build and support. Reasons include ambiguities in some information domains that make it difficult to decide precisely what to include in a compound, the question of whether a compound should be static or grow with the underlying information, the question of whether compounds should contain copies or references to their component information pieces, and the issue of how to enable people to create meaningful compounds without substantial overhead.

An example of compound building was found in the external technical support center, where several pieces of information from different sources frequently needed to be combined to solve a customer's problem. In addition to satisfying a particular customer quickly, a major piece of the support engineer's job was to ensure that future problems of the same or similar nature would be solved faster. One of the engineers described how he had solved a problem in the past from database records of customer calls:

> I found, say, four or five calls that really explain [some given problem] well from a couple of different angles. You know, I doubt if you'll find the perfect one [entry] that answers everything. You're likely to find a cluster of ones that, in the aggregate, provided good summarization.

Sometime later, this engineer needed to answer the same question again and had forgotten the answer. How could this situation be improved? There are two ways to leave a record of a problem: first, the engineer can create a new database entry which relates a particular problem to its solution. This will work well when the *exact* problem recurs. In this case, an information tool simply needs to allow easy access to the one new data item. But this approach loses information in two ways: another engineer can no longer learn anything about the first engineer's process of solving the problem, and the solution of a *related* problem requires a reconstruction of the information compound that represents the solution.

The second way of making future problem-solving more effective is to store the compound referentially, preserving links to original information that was used to solve the problem. Note, however, that the decision of what should be part of the compound requires thought. Let us look at a more complicated example.

A customer asked whether there was any software that would allow him to run a particular tape drive from some application. A search in the knowledge base revealed that a software patch existed to make this possible. A second search, over the patch database, produced the patch number needed to order the relevant supplementary software. The description of the patch, however, was misleading in one place and the support engineer contacted the "patch coordinator," a person specializing in the recording, maintenance, and shipping of software patches, to resolve the ambiguity. Satisfied, the engineer then caused the patch to be shipped to the customer, reminding himself to call him some days later to confirm arrival.

After this process we have an information compound with the following components: the original entry in the knowledge base which points to the patch, the patch description in the patch database, the ambiguity in the patch description (stored in the support engineer's head), the oral clarification from the patch coordinator, a record of the shipping request, and the reminder for the call-back. Which of these bits of information is important? Which are reusable? Which information management structures could best capture the important information?

It would certainly be wasteful to force another engineer handling the same question from another customer to go through the same search process. This would suggest the creation of a new knowledge base entry that includes all the information. But what if the second engineer tries to use the original patch pointer in the knowledge database, instead of the new entry? In this case, should the clarification from the patch coordinator be associated with the misleading entry in the patch database? These considerations suggest that each piece of the solution should be a separate entry in the information system. Separate compound entries would then reference the relevant solution entries for maximum flexibility in selecting the appropriate compound components.

The reminder to call back for confirmation should not be part of a permanent compound. Should that be kept in some separate space administered by the engineer until the information is irrelevant?

This example shows that information compounding is very important and by no means trivial. Note that the problem lies not with missing technology but with ambiguities in the domain. Tools intended to solve the problems of this engineer and his organization must take into account the fact that he is under time pressure. Also, both problem-solving strategies and level of patience for structural intricacies vary from person to person. We will return to this issue in a discussion of personal differences in Section 2.3.

An unusual form of compound building took place in the chemical analysis laboratory. When chemists encountered problems, they would have two different instruments analyze the same sample, each using its own method. The two results would then be displayed on the screen for comparison. In the case of drug production quality control, expected results, such as reference mass spectrometry patterns, were kept on paper for legal reasons. Compound construction support for information workers in this and other domains must therefore be able to include references to such paper-based components in its representation and maintenance strategies. More generally, components of compounds do not always come from a closed, on-line world. Important information might be available only in an inaccessible medium[2] or a temporarily unreachable data source.

2.2.2 Maintaining Compounds

Given the importance of information compounds, the question of maintenance needs to be addressed. It is difficult even to maintain primary data. Maintaining the secondary information represented by compounds is, of course, even more of an issue, especially since potentially different sets of people are involved in the two activities.

One problem in maintaining compounds is referential integrity. When entries referenced by a compound move or are deleted, the compound loses value. This problem is very visible on the World-Wide Web where links are frequently stale. For information stored completely under the control of one organization, deletion triggers available in commercial database systems can be used. For open systems like the Web, regular active integrity checks are needed. Of course, even when a referenced entry disappears, there may still be value in retaining the compound that contained it.

Apart from referential integrity of compound components, the lifetime of compounds is an issue. While some information compounds can be discarded after use, that is not true for regulated activities, such as drug quality control. All the compounds and operations on them must be recorded and retrievable[3].

When compounds are stored for future reference, determining the proper criteria for maintenance correctness are not always straightforward. Sometimes a particular information compound illustrates a problem and its solution and should therefore stay unchanged forever. At other times, however, the extent of a compound should stay flexible. An evolving sequence of mail messages which documents the solution to a problem can serve as an example. Whenever someone recalls this compound of messages, the result should include messages sent after the compound was defined. Similarly, when the disambiguation to an existing patch entry in the knowledge base was entered in the example above, it would have been useful to alert any future users who find the original patch entry of the new, clarifying information. Unless the information storage system has special capabilities, this would require modification of the old entry. That, however, may not be acceptable for legal audit, efficiency or quality assurance reasons. Tools would therefore need to enable non-intrusive 'attachment' of new information to old information, or to provide for flexible versioning. In general, whether to create static or dynamic compounds will depend on context, and compounding support tools need to be flexible in this regard.

Not surprisingly, technical issues were not the only barrier we found in the way of successful information maintenance. Reward structures are often not in place to encourage people in one group to create or maintain information that is primarily used by another group.

In the technical support group, we saw a characteristic example of this problem. Information of great importance to support engineers was available only if the field engineers who collected it had time to write it down. The field engineers themselves derived little benefit from the data. The lack of current information was a constant problem for the support engineers, but it was one over which neither they nor their immediate managers had control. Similar problems were reported in (Bowker and Star, 1991) where requests for very fine-grained categorization of the causes of death met with resistance from practicing physicians whose main interest was to proceed to the next living patient. Similarly, (Grudin, 1988) points out that the costs to non-benefiting parties is a source of resistance to cooperation in several groupware applications.

So far, we have mostly focused on the technical difficulties of structuring information. The more heterogeneous a cooperative arrangement becomes in terms of the people involved, the more problematic temperamental differences in working style become for the construction of generic tools. We now highlight some of these to exemplify tool requirements related to user temperament.

2.3 Working Style Differences: Detectives and Librarians

Some differences in the use of on-line information systems stem from skill level differences in the areas of general cognitive facility, such as the techniques of deduction or induction, expertise in the domain over which search is conducted, expertise around the computer system being used, or information seeking expertise, such as the use of indexes (Marchionini, 1995).

We also found marked temperamental differences in the approaches and attitudes of equally experienced workers in their search for information. In the technical support group, problem-solving style differences were recognized by both management and technical staff, and tolerance for differences was encouraged.

(Malone, 1983) describes how people have different needs for organization in their personal information spaces. A technical support engineer described some colleagues with a style different from his own:

> The person who is a little bit more of this kind of free agent, if you look at their desk, it's kind of this wilderness area—free association. Stuff's all over the place, but they get the answers quick.

He characterized two of the styles as “librarians” and “detectives.” Librarians are methodical and procedure-oriented. They generally enjoy or at least tolerate the activity of recording their actions on-line for future use by others, since they enjoy the teaching aspects of their jobs. Librarians can give a concrete description of their source preferences and recording habits, and they have the potential to be sophisticated information consumers and producers.

Detectives have a short attention span and are intuitive and impatient. They tend to dislike fine-grained information structure, preferring instead a few gross categories (Berlin et al., 1993). They follow hunches and “poke” the information sources. Their tolerance for documenting their actions is low. There is often a mismatch between a detective's thought processes and the recording process. Since detectives are intuitive, they may not be able to produce a logical chain of actions to write down. In addition, when a specific problem is solved, they lose interest, even if the solution could be generalized with additional work.

While these two stereotypes seem to reflect natural inclinations, some development of one person from one style to the other apparently does take place. One engineer told us that in the technical areas in which he was less experienced he had started as a 'librarian', but he had moved to a detective style as he gained expertise.

At first glance, the detective seems to be an informational parasite on the librarian. But an organization needs both kinds of specialists for different problems. Some problems require a leap of intuition to solve; others can be solved only by thorough and careful analysis. In the technical support group, collaboration and consultation were not uncommon. Engineers took advantage of style differences to ask for advice on calls where they seemed to be encountering communication or conceptual problems. It is part of the job of the information system designer and the surrounding organizational policies to support symbiotic, rather than parasitic, relationships between people of different working styles.

An immediately obvious place for an information system designer to cover both the detective and librarian stereotypes is in addressing the differences in patience levels. For example, a fine categorization of the causes of customer problems would allow very interesting analysis of these problems. While a librarian might be willing and interested in distinguishing between 'a user's misconception' and 'lack of information' when categorizing a problem, a detective would probably prefer to see the problem as simply 'pilot error'. For information input, a tool's willingness to accept less than fully categorized information will, of course, generally lead to some degradation of the overall system. But if the system were, for example, to insist on fine categorization, acceptance among the detective population would suffer and the system as a whole could be much more adversely affected.

For information display and browsing, a tool can cater to differences in temperament without detrimental effects at all. It could, for example, combine subcategories to present a coarser structure. Note therefore that while Section 2.1 showed the need for restructuring information for multiple uses, such restructuring needs also to be possible for reasons of different user temperaments.

Another place where tool design can help is in boosting the detective's tolerance for documenting for others. Automatic logging of some of the detective's work could alleviate the tedium of copying information from a work area into a call history report. A simple feedback mechanism could summarize especially for a detective information producer how much his work is being used by others. This would demonstrate tangible evidence of return on investment and might increase his willingness to expend the effort needed to create information for group use.

3. Information Use in and Among Tightly Coupled Groups

We now turn from the loosely coupled, amorphous, changing settings of the previous section to situations where tightly coupled, well-defined groups need to operate effectively within themselves and among each other.

(Greif and Sarin, 1987) discuss the technological support needed for data sharing within groups, dividing this support into two categories: data abstraction and control over sharing. The data management techniques they discuss, such as links, associative access, triggers, access control, and concurrency control, are extremely important elements of an information infrastructure for group work. However, they concentrate primarily on the situation of common data shared among a single group of individuals, each of whose needs must be met. Examples of data are shared calendars and multi-author documents. In these cases, there is a shared understanding of what the fundamental information objects are and what they mean.

In most of our domains there were certainly examples of within-group sharing of well-defined data, but in addition, cross-functional and cross-group sharing were common. We found that information shared among members of one group was different from what was shared across group boundaries. In the first section below, we describe these differences.

In the following section we describe some of the barriers we found standing in the way of sharing. Some of these are social, others are natural consequences of cross-group differences, and still others are of purely technical nature.

The last section, finally, describes the role of information and resource broker we found in several of the settings. These are individuals who broker contacts among members of various groups.

3.1 Inside vs. Outside: Different Habits and Materials

What kinds of information do people typically want to share? It often depends on whether the sharing occurs within an *insider* or *outsider* community. The feeling of being an insider community in our domains seemed to depend on having a common job or sharing a large piece of work and developing a sense of camaraderie. Insiders were not necessarily defined as a group of people who worked under the same manager, though often this was the case. In technical support, engineers certainly had the closest relationships with those who supported the same product lines and worked in the same groups, but they also seemed to feel a real sense of community with everybody at their site with the same job. In marketing, one of our informants felt that his closest coworkers were the people in scattered parts of the organization who managed similar products, rather than the people who were closest to him in the organizational tree.

As we will see, the information people shared with insiders was full of rich context and shared understanding of work. It described processes, changes, and rationale; these descriptions contained many interwoven threads, rather than isolated facts. Information shared with insiders was often a dynamic source, modeling current work. It was used to help people coordinate closely related activities, reach consensus, problem-solve, and teach.

For example, when the internal support group instructed their clients, or exchanged information with each other, available system software, network accessibility, the types of machines used, or the ability to obtain new hardware or software easily, were implicitly assumed and did not need to be mentioned. (Suchman, 1993) and (Goodwin and Goodwin) similarly describe how personnel in the ground operations of a major airline use interwoven information to 'read a scene', that is to develop an understanding of the current status of elements such as airplanes, baggage processing and passenger transportation, and how they work together around such information to ensure smooth operations.

The information people shared with outsiders tended to be simpler. It was used to describe interfaces between one work group and another, to convey an overview to outsiders such as the current status of a piece of work, and to provide isolated facts, such as names of contacts with whom to follow-up for more detail.

For example, it sometimes happened in the customer support group that consultation with sister groups supporting other products became necessary. For that purpose, the 'inside' material that included urgency, past problem history, customer contact person and more was distilled only to include the salient technical points to save the receivers time.

We now describe one particular kind of information we saw shared among insiders, and the resulting challenges for information tools. We close the section with observations about the problems with sharing beyond insider settings.

3.1.1 The Problem of Insider-Level 'How-To' Information

Product manuals or home repair books are typical examples of 'how-to' information shared with a wide audience. We found in our settings that insiders also attempted to share how-to information. But their much higher ambitions around this kind of information led to frustration.

Consider for example the information shared by workers in the software integration group. This group had difficult coordination tasks, both with their developer partners all over the company and within their own group, since several large operating system integration cycles were conducted in parallel and managed by different group members. The information artifacts in this group were about processes. In essence, each was a template describing a *prototypical* process, or 'how-to' information.

Specialized expertise and roles regarding these processes had developed informally, yet each person also rotated through the position of "wizard of the week," in which questions of outsiders (component developers or others) were answered on behalf of the group. The integration group needed process descriptions both to coordinate their work and to enable them to wear one another's hats. This level of information proved difficult to produce. A member of the integration group described a frustrating scenario:

> In each area of expertise, there are things that everybody knows—the tricks of the trade... I have watched the other person do this a million times, but I've never actually sat down and done it and I was running it late Tuesday night, and I ran it and [it] completely skipped one section. So I ran it again and it skipped the section again, and that was the section I really wanted to take care of, and so of course the person wasn't there because it was midnight, and it turned out I had to touch a particular file before, so its date would be more recent than the

> most recent posting. So there's all these little hidden pieces of information.
>
> We have a handbook that we tried to write, but quite frankly, for me to sit down and read this handbook, I already know 90% of the information that's in the handbook, and so for me to sit down and tediously read every word in this handbook, I don't think to myself that there would be a very high return on that, but somewhere hidden in there is probably this piece of information that you have to touch this file and that there's probably a very good reason why he has the code there.

This is how the author of the handbook described a new person in the group using the handbook:

> Even if she did have it on-line she'd have to know a lot to find the information that explains how to... because you could use the output of the system.group[4] for a number of different things and so I don't list every one, you know, I don't list, "to find out whether or not the partner can access the database you would do *this*," because that's one of twenty things you might do with that particular piece of information. The information is sort of there, but it would take a persistent person to actually find all the information to apply it—partly because it's scattered, partly because no individual problem is identified.

The complexity of the how-to information attempted to be shared here poses difficult challenges for information tools that try to help.

Process-related information such as the software integration handbook or the customer call logs described earlier seems to have three basic components: *context*, *control flow*, and *exceptions*. The context identifies the current place in the process well enough to make the data understandable and to evoke the proper questions and structures in the user's mind. In call histories, the context included a description of the customer's hardware and software environment, the state of the system when the problem occurred, the symptoms of the problem, and any diagnostics or repairs that had already been attempted.

Control flow information produces the choices of actions at a particular point. Continuing our example, this would include tests that might help refine the problem, documentation to consult, and commands that might change the state of the system.

Exceptions in process-related information convey how the current step might fail and how it can be recovered. For instance, running a repair script on a mangled database might result in a loss of data; in this case, returning to an earlier version might be the only response, or it might be possible to salvage some of the data that remained.

The representation of these three process information elements must fill a dual purpose. It must allow people to learn or observe the entire process by extracting overviews efficiently. At the same time it must produce specific information when the process goes wrong or when decision points are reached. Which information will need to be extracted cannot be known in advance, so the process description must be complete enough to allow users to “tap in” at many different points. For example, when a software installation encounters a problem in midstream, and the user finds the place in the process description that corresponds to the current state of the installation, it would be useful to get a summary of assumptions (current context) supposedly established by previous installation steps, as well as procedures for testing the validity of these assumptions. These were the ambitions we found expressed by the designers of insider information.

In the context of examining expert help systems, (Suchman, 1987) points to some of the subtleties involved in the (indirect) communication between instructors and someone following their directives. She observes that: “...instructions rely upon the recipient's ability to do the implicit work of anchoring descriptions to concrete objects and actions (p101)”, which is in part what the engineer quoted above is trying to do. Suchman further observes that

> More than the “correct” execution of an instruction, in other words, successful instruction-following is a matter of constructing a particular course of action that is accountable to the general description that the instruction provides. The work of constructing the course is neither exhaustively enumerated in the description, nor completely captured by a retrospective account of what was done. (p102)

Each of the three components of a process description is necessary for the description to be effective, yet the natural storage and presentation strategies for each of these components are quite different. To add to the difficulty, there are complicated interrelationships among the three components. Context, for example, suggests a summary of which conditions hold at a particular point. Context is cumulative, and the contextual data associated with any one step is probably not self-contained, but depends on the path taken so far in the process. Control flow suggests a filtered view of a particular thread from a complex array of possible branch points. Exceptions suggest a rule-like format, tying together elements from both context and control flow.

3.1.2 Sharing With Outsiders

Information shared with outsiders was quite a bit simpler to manage. In technical support, for example, engineers used patch and bug databases to help them when solving problems. These were narrow status reports, giving only brief descriptions, important dates, and access information. This information still reflected work interdependencies, but the brevity and simplicity of the information indicated a more distant relationship between the workers, with less need or ability to influence one another's actions.

Of course, the relative simplicity of the shared information does not mean it was easy to produce. This was made clear by examples when the ownership of a customer problem had to migrate from one customer support group to another. This occurred when engineers believed that the second group had more expertise in the problem area in question. The transfer effectively required the transformation of an 'insider corpus' into an outsider corpus of information. Though the engineers shared a common culture and set of information tools across groups, greater care was taken in managing such a transfer than in a close collaboration between team members. A semi-formal protocol helped people decide which information should be transmitted, to make the transfer as efficient as possible. Care was taken to avoid overloading or wasting the time of the recipient engineer.

(Strauss et al., 1985) describe similar observations in hospital settings, where handoffs occur both across shifts of workers concerned with the same aspect of patient care, and among teams concerned with different nodes on one 'task trajectory', a plan of action comprising a patient's treatment. They describe how a mixture of written and oral transmission of information is used, and how this information is sometimes enriched with personal observations and warnings about particular patients' idiosyncrasies.

This kind of information transformation usually takes intellectual work. But when written records are required, information compounds again come into play. The body of information to be produced in the transformation is in part an excerpt of the insider corpus. New information may need to be added to make up for implicitly assumed knowledge that is not in the original information. An example is product design rationale that is assumed to be understood in the context of the insider group. That might, for instance, be the fact that a new product under examination is intended to be especially low-cost, or particularly strong in performance. Tools supporting extraction and summarization for the purpose of moving information across group boundaries would have been important in several of the settings we visited.

In the product development groups, there was an interesting variation on the insider/outsider notion. In this setting, the boundaries of community were sometimes fluid. Parts vendors seemed to be clearly outsiders, yet they were in some cases working closely in collaboration with manufacturing process engineers inside the company. It would have been very convenient to directly provide vendors with on-line CAD models instead of drawings generated from them, especially when iterations over the design and manufacturing processes were necessary. However, the tool technologies of the two domains were not the same, preventing such sharing from taking place.

3.2 Barriers to Dissemination

Incompatibility of tools is a common difficulty in the way of sharing, but there are others we found in the settings we visited.

3.2.1 Organization of Groups

Some groups structure their work so each member has an independent task to work on, unrelated to the tasks of his or her coworkers. For example, in the finance group, the vendor accounts were split alphabetically among group members. There was little collaborative work and little motivation or need to share information about vendors in this setting. Although it would have been useful to share information about the different billing systems that each group member was likely to encounter, the alphabetical partitioning of work did not permit one person to easily learn who had expertise with which systems.

Other groups set up independent but interrelated activities for each person. For example, in product development it is expected that many highly interdependent activities will proceed in parallel. In the software integration group, an evolving test system was continually going through a limited integration process at the same time a major release was being prepared. The information flow needed by groups like these is difficult to coordinate; access to partial or incomplete data is often needed. In addition, the issues of who may update the interrelated information and how others are informed of updates are critical. Explicit policies are often needed, with accompanying support for access controls and update notification from the information system.

In addition to the organizational setup of groups, other evidence shows that the duration of a group as a unit affects information flow. (Katz and Allen, 1982) show that, at least in research and development project groups, the willingness to *accept* incoming information declines as groups age. Such resistance to information must be reckoned with as well when evaluating the effectiveness of information dissemination tools.

3.2.2 Trust

In the groups we studied, sharing of information most readily occurred within a small community in an informal manner. A marketing engineer would share a rumor with a group of coworkers but would not post a note about it to a bulletin board. One reason is clear; trust exists between a group of insiders that makes it easy to risk passing on information that may not be correct, as long as it has sufficient potential value. For example, a support engineer would share a useful computer script within his team without worrying about whether it was completely robust.

Sometimes the “in progress” nature of information is a barrier to its dissemination. Although incomplete information is less reliable than information which has reached a stable state, it can be very valuable when it is shared.

In technical support, histories of customer calls were not generally accessible to other support centers until the problem had been solved and the call had been closed. It was usually impossible to find out when two calls about the same problem were in progress at different sites, which inhibited the problem-solving process. We saw an exception to this rule when a manager from one support location happened to be visiting another location and saw an engineer working on a problem that he knew his own group was trying to solve. This kind of serendipitous discovery was considered rare but very valuable.

Making changing information visible contradicts the traditional database transaction model. If information systems are to promote this kind of information sharing, database developers will have to broaden their transaction models to permit views of changing information. Different levels of sharing of incomplete information may be desirable in different domains. In the call history example, it might be enough to know that a call about the same problem is in progress rather than see the contents of the uncompleted call; the next step might be to contact the other engineer and coordinate problem-solving efforts. In marketing, information about products and competitors continually evolved, and snapshots of the contents of changing data would be best.

When the dissemination of incomplete, preliminary information is limited by trust, rather than by technical considerations such as data consistency, tools could allow users to specify “circles of trust” over the user community and could allow information to carry reliability and stability attributes. For example, rather than the all-or-nobody visibility rules of conservative database transaction management schemes, particular computer processes, domains, or users could be allowed to view preliminary information, maybe following an optimistic concurrency control scheme for those cases, or even just allowing read access to potentially unstable data. See (Greenberg and Marwood, 1994) for some further analysis and suggestions in this area.

3.2.3 Technical and Organizational Overhead

The overhead of disseminating information can be a barrier to sharing it. Some people offer information verbally in a group meeting but do not feel they have time to write it down for others. In one case, a technical support person made copies of a printout of some useful information to distribute at a meeting, rather than sending it through electronic mail, because he felt that the effort of maintaining an appropriate distribution list was too great. Information systems can facilitate sharing by providing low-overhead export mechanisms.

In some settings, management approval may be necessary before sharing information. This was most apparent in marketing, where there was an explicit approval mechanism in place for giving information to people outside the work group. Knowledge is an asset; people do not always want to share what they know. Much of the marketing information was confidential, even within the company. In addition, we encountered managers who did not want the people they managed to advertise their expertise, since that might bring them an undesirable work load.

Terminology can also be a problem. Within a group, terminology is usually well understood, but it is still far from standard or unique. In the software integration group, different abbreviation and naming conventions were used to refer to the same software components and processes, which made it difficult for people to use the rich databases they had created. Thesauri can be used to help bridge the gap between people with different backgrounds or levels of expertise. Thesauri can be user-customizable and can evolve along with the domain they describe.

Their current use is mostly to facilitate knowledge-based retrieval by enlarging term sets. One suggestion would be to extend their use to support sharing and communication within groups. For more discussion of thesauri in information retrieval, see (Hoppe et al., 1990; Kracker, 1990).

3.3 Contact Brokers

The "terseness" and "unfriendliness" of computers sometimes seems to be especially apparent when on-line information is shared among diverse groups of people. Therefore, other people are often chosen as information sources. This can sometimes be difficult.

As an example, let us return to the product development engineer designing a new paper feed. The old project documentation and notebooks consulted by the engineer constituted a shared resource, created as a by-product of the former product development processes. To make sense of the material, the engineer had to understand the context of these old designs—the priorities used in decision-making, which designs were rejected and why, and the changes that were made during the design process. This kind of complex, interwoven information is extremely difficult to record, especially on behalf of people outside the work group, yet it is precisely what is needed by anybody interested in design reuse. In this scenario, the engineer used the existing information to understand the background to the problem. An important step was then to call on the designers who were still with the company to guide him through the material and provide additional context. Such auxiliary use of intermediaries can be a very successful strategy for dealing with shared information resources. In fact, experts need to be consulted in the context of current projects as well. The problem is that often individuals in need of advice do not know where to turn beyond their own tightly coupled group.

The role of contact broker has emerged to address this problems in the various settings we visited. These brokers help group members find experts in other groups. While some managerial positions explicitly called for such activities, the contact brokers we saw in non-managerial positions had developed informally.

We saw an example in the internal technical support center, where brokers did not solve all problems themselves but were aware of activities and expertise of others and could direct clients to those experts:

> Our expertise is both global and individual. "Oh, I think Jeff works on mathematical type stuff and he's an EE and so he would probably know if not himself, who to speak with." And so I think a lot of our knowledge is who to go to and where to go, much like a reference librarian. And [about] *that*, we have some written down, but very, very little.

Much more was happening in this case than simple matching between problems and experts. Depending on the level of trust between the technical support broker and a client, the broker would provide additional information, such as possible biases of a particular expert. Experts were willing to have the broker send clients to them for help because the broker exercised judgment over how many and which kind of clients he sent. The broker would, for instance, be careful to send novices to experts with an interest in teaching or at least some tolerance for naive questions.

The broker *deduced* which person had particular expertise or interests from his monitoring of electronic mail distributions and bulletin boards and from questions asked of him by others. He also used his understanding of the problem domains and the organization to make likely assumptions: "If someone in the user interface group built an on-line telephone interface, she might know about buttons in a Motif interface." In environments with a large number of heterogeneous sources, this kind of expertise in discovering and selecting sources is invaluable and can greatly facilitate the sharing of information across tightly couple groups.

The role of information brokers is distinct from that of *gatekeepers* (Tushman and Katz, 1980). These are individuals who are locally associated with the group they serve and are intimately involved in its tasks. In addition, they maintain strong ties to other projects and act as mediators of the information flow across group boundaries for other members of their working group. This mediation process includes translation of the information into terms the more locally oriented group members can understand. (Allen, 1977) also studied the phenomenon of gatekeepers in research and development organizations as individuals who excel in maintaining outside contacts from which they funnel information into their organization, again, acting as conduits for the information flow. He stresses the importance of their role for the health of their organizations.

In contrast, the information brokers among our informants were individuals who brokered *contacts* between groups other than the one they were organizationally part of, as well as providing information directly themselves. They were, however, highly respected within their own work groups, differentiating them as well from individuals with merely representational roles who also sometimes act as contact brokers.

We conjecture that a formal on-line ‘brokerage’ tool would not work well, because the role of expert is often informal, with assistance to others given unofficially. Due to this informality and the delicate balance between job-related work and assistance function, we suspect that potential candidates would shy away from a formal expert sign-up mechanism which implies obligation and lacks protection against overuse, a function our human brokers performed. It is also worth noting that contact brokers evolved in addition to existing organizational measures designed to foster cross-group interaction. These included brown-bag lunches, beer busts, company picnics and off-site retreats.

4. Conclusion

The World-Wide Web has dramatically increased the number of individual publishers of information, and it has made it technically easier to access on-line material of varying formats. It has thereby smoothed the physical information pathways between information providers and consumers. It has also brought basic hypertext structuring technology into large-scale use. The evolving Java work promises to add technology for delivering attractive information display capabilities and interactivity together with the information itself . But the Web by itself cannot solve the larger cooperative work problem.

Our interviews brought out some of the deficiencies in the current level of technical and organizational support for information handling. Few of these are addressed adequately in widely available technologies. To address these deficiencies, it is essential to understand the real-world complexity of managing information in the technical workplace, to build tools in situated designs (Greenbaum and Kyng, 1991). This complexity is revealed in the variety of interrelated information sources, information-intensive tasks, and scenarios of information use.

Information sources can be on-line or off-line, written or verbal, broadcast or personal, structured or unstructured. The same information is often used in multiple settings for different purposes. An individual must master a large number of retrieval and interaction mechanisms to access the variety of sources needed for many tasks.

In many technical settings, the flow of information is the basic nourishment needed to accomplish large tasks. Information is needed to coordinate efforts within and between groups, establish and monitor processes, and inform decisions. Such information can have a tightly interwoven structure, making it difficult to organize and search. Information is sometimes shared among tens or hundreds of people with different backgrounds and expertise, which requires extra care in choosing its representation, access methods, and views.

Scenarios of information use include the creation of information compounds, which filter information from different sources and assemble the resulting information components into a useful view for a particular task context. Different styles of information access are preferred by people with different approaches to problem-solving or decision-making, such as ‘detectives’ and ‘librarians’.

Both technology and the organization of work must play a role in dealing with the complexity we have described. The technological changes needed include support for source integration, retrieval and production mechanisms that allow the creation and reuse of information compounds, flexible mechanisms for disseminating and updating shared information, and new techniques for structuring interwoven information such as design plans or problem-solving histories.

Organizational support is needed to reward tool champions, encourage information brokers, motivate source maintenance and dissemination, and in general to allow work processes to evolve as changing technologies bring new capabilities to the workplace.

Throughout this article, we have described scenarios from our interviews which illustrate people's need to personalize shared information and to share individual information. This migration of information in and out of different information spaces is constant, as information is created, imported, and moved from one individual or group to another. To accommodate frequent transformations in the information geography, information tools and the organizational contexts must be designed as enablers, not inhibitors, of change.

Historically, traditional data processing operations have primarily supported corporate decision makers and specialty clerks, such as inventory managers. As more information becomes available and indispensable to much broader sections of organizations, information systems must be redesigned to serve this large and varied collection of people.

5. Appendix: Observations for Tool Designers

As we analyzed our interview data, we found certain clusters of information categories or themes that helped us organize what we had learned. These themes and their related issues, which we present below in a tabular format, represent different aspects of information as considered in the context of the workplace and workers' activities. Each theme suggests a whole host of design questions for information tool designers. The tables below are not intended to be an interview or observation plan—they are not at the right level for that purpose. Instead, they capture a set of issues and domain forces for tool designers to use as a guide while trying to understand their own application domains.

Each table is divided into three columns. The first identifies a cluster of issues. The second lists a sample of factors to examine in the context of the cluster. The third lists a few references that may be helpful. These are, of course, only starting points. They are not exhaustive bibliographies.

Table I lists issues around the information sources used in a setting. Table II brings up clusters to explore in regards to information contents. Table III suggests topics centered around information usage. Table IV continues with the problem of information maintenance, while Table V, finally, lists clusters concerned with organizational issues.

Table I: Information Sources

Areas for Exploration	Related Factors	Literature
Which information is available on-line—why?	Media, location, limitations, lifetime, interoperability, reliability, accessibility, security.	(Davis, 1990; Gupta, 1989)
Which information comes from conversations—why?	Brokers, specialists, subtle interaction conventions.	(Tushman and Katz, 1980; Allen, 1977; Cicourel, 1990)
Which information comes from paper—why?	Legal restrictions, convenience, comfort, mobility.	(Harper and Hughes, 1993)
How are task handoffs managed?	Formal and informal guidelines, translation tools.	(Strauss et al., 1985)

Table II: Information Contents

Areas for Exploration	Related Factors	Literature
How is information structured?	Free text, regular/ irregular, consider users' internal models.	(Conklin, 1987; Shepherd, Mayer, and Kuchinsky, 1990)
Is information structure changing?	Who needs to change it? Under what circumstances?	(Röscheisen, Mogensen, and Winograd, 1995)
Is interwoven information stored and used?	Process, histories, rationale, constraints, context.	(Suchman, 1993; Suchman, 1987; Goodwin and Goodwin)

Table III: Information Usage

Areas for Exploration	Related Factors	Literature
Are compounds used?	Static or live, to be saved or disposable.	(Halasz, 1988; Motro, DAtri,andTarantino,1988;Levy,1994)'
Are ad hoc or imprecise queries used?	Look for retrieval patterns.	(Zloof, 1977; Kim, 1986; Burgess, 1991; Fischer and Nieper-Lemke, 1989)
Is retrieval expertise identifiable?	Find specialists, macros, softkeys or scripts being used.	(Wiederhold, 1992)
Is browsing done?	Large search spaces? Can structure be used for support?	(Business Periodicals Ondisc Users Guide, 1989; McAleese and Duncan, 1987; Egan et al., 1989; Ackerman and Malone, 1990; Caplinger, 1986)
Are multiple views and terminology used?	Check professional roles of multiple users, cross-check usage in multiple contexts.	(Dumais et al., 1988; Bowker and Star, 1991)
Is information archived or moving?	How do users cope with monitoring information?	(Yan and Garcia-Molina, 1995)
Should sources be active?	Information 'falling through the crack', need for coordination of work, data consistency problems.	(Risch, 1989)
Is information shared?	What are the groups. Who are the insiders and outsiders?	(Hoppe et al., 1990; Kracker, 1990; Dumais et al., 1988)
What are current barriers to sharing?	Diverse population, practical obstacles, approval requirements, user resistance.	(Katz and Allen, 1982)

Table III: Information Usage

Areas for Exploration	Related Factors	Literature
Are heterogeneous sources in use?	Discovery/selection/ interaction mechanisms.	(Gravano and Garcia-Molina, 1995)

Table IV: Information Maintenance

Areas for Exploration	Related Factors	Literature
Is information deleted?	Legal restrictions, resource restrictions, comfort level.	
Is information modified?	Legal restrictions, dependency maintenance, motivational issues.	(Risch, 1989; Greenberg and Marwood, 1994)

Table V: Organizational Issues

Areas for Exploration	Related Factors	Literature
The 'official' work process	Identify different viewpoints. Is work event-driven or planned?	
The actual work process. Deviations from the official process.	Coordination and interaction mechanisms. Shortcuts, tolerated exceptions, temperamental differences, power plays.	(Flores et al., 1988; Kaplan, 1990; Strauss et al., 1985)
Process breakdowns.	Missing information. Is the process fundamentally flawed or just poorly supported?	(Strauss et al., 1985)
Skill distribution.	Dependencies between skills and tools. Example: how might an information system change skill distribution?	(Kraut, Dumais, and Koch, 1989)

Table V: Organizational Issues

Areas for Exploration	Related Factors	Literature
Individual differences in temperament and work habits.	Speed, search techniques, communication patterns. Librarians/detectives.	(Schiele and Hoppe, 1990; Wilson and Herot, 1980)
Formal and informal specialist roles and how they are used.	Organizers, tool usage innovators. How and why workers access the services of specialists.	(Nardi, 1993; Nardi and Miller, 1991; Tushman and Katz, 1980)
Turnover rates.	Will tools be used by frequently changing personnel? Mostly novice users?	(Marchionini, 1995)
Characteristics of novices and experts.	Learning times, particular learning problems, teaching techniques used.	(Goodwin, 1991)
Reward system.	What do people find satisfying? What is valued and what is not?	(Bowker and Star, 1991; Grudin, 1988)
Metrics.	How are metrics used in the work process? How are they perceived?	

6. References

Mark S. Ackerman and Thomas W. Malone (1990). Answer Garden: A Tool for Growing Organizational Memory. In *Proceedings of the Conference on Office Information Systems*, pp. 31-39.

Thomas J. Allen (1977). *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*. MIT Press.

Lucy M. Berlin, Robin Jeffries, Vicki L. O'Day, Andreas Paepcke, and Cathleen Wharton (1993). Where Did You Put It? Issues in the Design and Use of a Group Memory. In *INTERCHI '93*, pp. 23-30.

Geoffrey Bowker and Susan Leigh Star (1991). Situations vs. Standards in Long-Term, Wide-Scale Decision-Making: The Case of the International Classification of Diseases. In Jay F. Nunamaker, Jr., editor, *Proceedings of the Twenty-Fourth Annual Hawaii International Conference* *on System Sciences*, pp. 73-81. IEEE Computer Society Press.

Clifford G. Burgess (1991). A graphical, database-querying interface for casual, naive computer users. *Int. Journal Man-Machine Studies*, 34:23-47.

Business Periodicals Ondisc Users Guide (1989). University Microfilms International, 300 North Zeeb Rd., Ann Arbor, Michigan 48106.

Mary Campione and Kathy Walrath (1996). *The Java Language Tutorial*. Addison-Wesley. Draft available at http://java.sun.com/tutorial/ index.html.

Michael Caplinger (1986). Graphical Database Browsing. In *Proc. 3rd ACM-SIGOIS Conference on Office Information Systems*, pp. 113-121.

Aaron V. Cicourel (1990). The Integration of Distributed Knowledge in Collaborative Medical Diagnosis. In Robert E. Kraut and Carmen Egido Jolene Galegher, editors, *Intellectual Teamwork*, pp. 221-242. Lawrence Erlbaum Associates.

Jeff Conklin (1987). Hypertext: An Introduction and Survey. *IEEE Computer*:17-41.

J. Steve Davis (1990). Experimental investigation of the utility of data structure and E-R diagrams in database query. *Int. Journal Man-Machine Studies*, 32:449-459.

Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman (1988). Using Latent Semantic Analysis to Improve Access to Textual Information. In *Proceedings of the Conference on Human Factors in Computing Systems*.

Dennis E. Egan, Joel R. Remde, Louis M. Gomez, Thomas K. Landauer, Jennifer Eberhardt, and Carol C. Lochbaum (1989). Formative Design-Evaluation of SuperBook. *ACM Transactions on Information Systems*, 7(1):30-57.

Gerhard Fischer and Helga Nieper-Lemke (1989). HELGON: Extending the Retrieval by Reformulation Paradigm. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 357-362.

Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd (1988). Computer Systems and the Design of Organizational Interaction. *ACM Transactions on Information Systems*, 6(2):153-172.

Charles Goodwin (1991). Situated Literacy in the Workplace. Colloquium, UCLA Deparment of Applied Linguistics.

Charles Goodwin and Marjorie Harness Goodwin. Formulating Planes:

Seeing as a Situated Activity. In Y. Engestrom and D. Middleton, editors, *Communication and Cognition at Work*. Cambridge University Press. (in press).

Luis Gravano and Hector Garcia-Molina (1995). Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies. In *Proceedings of the Twenty-first International Conference on Very Large Databases*, pp. 78-89.

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

Saul Greenberg and David Marwood (1994). Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface. In Richard Furuta and Christine Neuwirth, editors, *Proceedings of the Conference on Computer-Supported Cooperative Work*. ACM.

Irene Greif and Sunil Sarin (1987). Data Sharing in Group Work. *ACM Transactions on Office Information Systems*, 5(2):187-211.

Jonathan Grudin (1988). Why Groupware Applications Fail: Problems in Design and Evaluation. *Office: Technology and People*, 4(3):245-264. An earlier version of this paper was published in the Proceedings of the Conference on Computer-Supported Cooperative Work, CSCW '88.

Amar Gupta (1989). *Integration of Information Systems: Bridging Heterogeneous Databases*. IEEE Press, New York. This is a collection of papers on the subject.

Frank G. Halasz (1988). Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM*, 31(7):836-852.

R.H.R. Harper and John A. Hughes (1993). 'What a F-ing System! Send 'Em All to the Same Place and then Expect us to Stop 'Em Hitting'. Making technology work in air traffic control. In Graham Button, editor, *Technology in Working Order. Studies of Work, Interaction, and Technology*. Routledge.

K. Holtzblatt and S. Jones (1990). *Contextual Inquiry: Principles and Practice*. Number DEC-TR 729. Digital Equipment Corporation.

Heinz Ulrich Hoppe, Karin Ammersbach, Barbara Lutes-Schaab, and Gaby Zinssmeister (1990). EXPRESS: An Experimental Interface for Factual Information Retrieval. In *Proceedings of the Thirteenth International Conference on Research and Development in Information Retrieval*,

pp. 63-82.
Simon M. Kaplan (1990). Conversationbuilder: An Open Architecture for Collaborative Work. In *Human-Computer Interaction - Interact '90*, pp. 917-922. Elsevier Science Publishers B.V. (North-Holland).
Ralph Katz and Thomas J. Allen (1982). Investigating the Not Invented Here syndrome: A look at the performance, tenure, and communication patterns of 50 R&D project groups. *R & D Management*, 12(1).
Hyoung-Joo Kim (1986). Graphical Interfaces for Database Systems: A Survey. In *Proc. of the 1986 Mountain Regional ACM Conference*.
Martin Kracker (1990). *Approximate Concept Matching Using a Fuzzy Network of Concepts*. Gesellschaft für Mathematik und Datenverarbeitung (GMD), Integrated Publication and Information Systems Institute.
Robert Kraut, Susan Dumais, and Susan Koch (1989). Computerization, Productivity, and Quality of Work-Life. *Communications of the ACM*, 32(2):220-238.
David M. Levy (1994). Document Reuse and Document Systems. *Electronic Publishing—Origination, Dissemination and Design*, 6(4):339-348.
Thomas W. Malone (1983). How do people organise their desks? Implications for the design of office information systems. *ACM Transactions on Information Systems*, 1(1):99-112.
Gary Marchionini (1995). *Information Seeking in Electronic Environments*. Cambridge Series on Human-Computer Interaction. Cambridge University Press.
R. McAleese and E.B. Duncan (1987). The Graphical Representation of 'Terrain' and 'Street' Knowledge in an Interface to a Database System. In *11th International Online Information Meeting*, Aberdeen University, UK.
Amihai Motro, Alessandro D'Atri, and Laura Tarantino (1988). The Design of KIVIEW: An Object-Oriented Browser. In *Proceedings of the 2nd International Conference on Expert Database Systems*, pp. 107-131.
Bonnie A. Nardi and James R. Miller (1990). An Ethnographic Study of Distributed Problem Solving in Spreadsheet Development. In *Proceedings of the Conference on Computer-Supported Cooperative Work*, pp. 197-208.
Bonnie A. Nardi, Heinrich Schwarz, Allan Kuchinsky, and Robert Leichner (1993). Turning Away from Talking Heads: The Use of Video-as-Data in

Neurosurgery. In *INTERCHI93*, pp. 327-334. ACM Press.
Bonnie A. Nardi and James R. Miller (1991). Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies*, 34(2):161-184.
Bonnie Nardi (1993). *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press.
Tore Risch (1989). Monitoring Database Objects. In *Proceedings of the Fifteenth International Conference on Very Large Databases*, pp. 445-453.
Martin Röscheisen, Christian Mogensen, and Terry Winograd (1995). Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities. In *Proceedings of the Third International World-Wide Web Conference*. For an expanded version of this paper, see http://www-pcd.stanford.edu/COMMENTOR/.
Franz Schiele and H. Ulrich Hoppe (1990). Inferring Task Structures from Interaction Protocols. In D. Diaper et al., editor, *Human-Computer Interaction - Interact '90*, pp. 567-572. Elsevier Science Publishers B.V. (North Holland).
Allan Shepherd, Niels Mayer, and Allan Kuchinsky (1990). Strudel: An Extensible Electronic Conversation Toolkit. In *Proceedings of the Conference on Computer Supported Cooperative Work*, pp. 93-104.
Anselm Strauss, Shizuko Fagerhaugh, Barbara Suczek, and Carolyn Wiener (1985). *Social Organization of Medical Work*. The University of Chicago Press.
Lucy Suchman (1993). Centers of Coordination: A case and some themes. In *Proceedings of the NATO Avanced Research Workshop on Discourse, Tools and Reasoning*, Lucca, Italy.
Lucy A. Suchman (1987). *Plans and Situated Actions. The Problem of Human Machine Communication*. Cambridge University Press.
Michael L. Tushman and Ralph Katz (1980). External Communication and Project Performance: an Investigation into the Role of Gatekeepers. *Management Science*, 26(11).
Gio Wiederhold (1992). Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3):38-49.
Gerald A. Wilson and Christopher F. Herot (1980). Semantics vs. Graphics—To Show or Not to Show. In *Proceedings of the Sixth International Conference on Very Large Databases*, pp. 183-196.

Dennis Wixon, Karen Holtzblatt, and Stephen Knox (1990). Contextual Design: An Emergent View of System Design. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 329-336.

Tak W. Yan and Hector Garcia-Molina (1995). SIFT—A Tool for Wide-Area Information Dissemination. In *USENIX Technical Conference*, pp. 177-186. http://sift.stanford.edu.

M.M. Zloof (1977). Query-by-Example: a data base language. *IBM Systems Journal*, 16(4):324-343.

7. Footnotes

1. In the context of hypertext nodes, compounds are similar to the "composites" outlined in (Halasz, 1988).

2. (Harper and Hughes, 1993) describe how air traffic controllers use paper-based "strips" in conjunction with computer systems for tracking airplane movement. They describe how the controllers manipulate these strips in a rich variety of ways to do their job. These include sorting, marking and highlighting through placement.

3. For an analysis of the role of information compounding in document reuse and the function of document interchange standards in this context, see (Levy, 1994).

4. 'System.group' is a technical jargon term used by the system integration team.

*Acknowledgments

The essence of this paper is the information related to us by the people we interviewed. In spite of their work loads, they spent hours demonstrating their activities and sharing their thoughts. In addition, Vicki O'Day made rich contributions in thought and action. Lucy Berlin, Lew Creary, Danielle Fafchamps, Robin Jeffries, Nancy Kendzierski, Bonnie Nardi, and James Navarro read early drafts of this paper and provided us with valuable insights.