We have interviewed information workers in widely differing areas of a large, diverse company. We describe some of the unsatisfied information needs we observed during this study. Two clusters of issues are described. The first covers how individuals use information, customizing it for particular tasks by assembling selected information from different sources into information compounds. The problems of identifying, selecting, and interacting with heterogeneous sources are discussed. The second set of issues centers around the characteristics of and barriers to sharing information among work groups. Effective sharing depends on factors such as trust, the ability to personalize, and the addition of meaningful structure to complex, interwoven information. We discuss the implications of these information needs for both technological and organizational change.
## Contents

1 Introduction ........................................... 3  
1.1 Methodology ........................................ 3

2 Individuals' Uses of Information ....................... 5  
2.1 Using Multiple Sources .............................. 6 
2.1.1 Identifying Sources .............................. 6 
2.1.2 Selecting Sources ................................ 7 
2.1.3 Interacting with Multiple Sources ............... 8
2.2 Custom Information Compounds ...................... 10 
2.2.1 Building Compounds .............................. 10 
2.2.2 Capturing Filtering and Retrieval Expertise ....... 12 
2.2.3 Structuring Information for Multiple Uses ....... 13

3 Information for Group Use ............................ 14 
3.1 Sharing Within Circles of Trust ................. 15 
3.2 Personalization of Information ................ 17 
3.2.1 Detectives and Librarians .................... 17 
3.2.2 Formal and Informal Roles .................... 18 
3.3 Interwoven Information ........................ 20 
3.4 Information Brokers .............................. 23 
3.5 Source Maintenance ............................... 24

4 Summary ............................................. 26

5 Conclusion ........................................... 29

6 Acknowledgments ...................................... 30

7 References ........................................... 31
1 Introduction

With high-speed computer networks and sophisticated storage technology in widespread use, information access may seem to be a solved problem. However, networks and technology address only the problem of physical access to data. Methods for dealing with information at the level needed by information workers lag far behind. People who manipulate information as an integral and significant part of their jobs more than ever need support for handling large amounts of diverse information with as little overhead as possible.

To better understand the needs and characteristics of information workers, we interviewed people working in widely differing areas within a diverse company. The information worlds used by our informants were surprisingly large and varied, supporting in most cases complex work processes. The real-life problems people encountered and the strategies they employed in dealing effectively with information raise important issues for designers of information technology and information-responsive organizations.

Without the perspective that comes from examining real-world activities and problems, it is easy for designers to oversimplify the requirements of new technologies, particularly for information systems. To solve information problems, designers must understand the rich context of current information usage, with human interactions taking place through and around information sources and information-based tasks. The movement toward situated design, as outlined by Greenbaum and Kyng in [1], suggests that much greater attention needs to be given to designing with the full participation of users, to create tools that enhance work processes and fit gracefully into the work environment.

Similarly, organizational scientists may concentrate on changes to processes and overlook the need for technical innovation. We believe that it is critical to understand both the technological and social aspects of information-related problems, since solutions in one arena are often handicapped by ignorance of or lack of progress in the other. Flores et al. [2] described technology as "the design of practices and possibilities to be realized through artifacts." That is, it is difficult to consider technological change separately from its social impact. In this paper, we show the implications of information needs in both areas.

We will discuss two clusters of information-related issues that arose from our study. The first cluster covers some of the ways in which information was put to use by individuals. A unifying theme of the study is that information was often customized for particular work contexts and tasks. This customization involved reusing information in different settings, applying filters to large data sources, and making retrieval choices when dealing with heterogeneous sources.

The second cluster of information-related issues covers how information was used by work groups to coordinate, collaborate, record group practices, and share results. The need to share information does not imply that it is easy to describe it for community use; there are many barriers both to making information sharable and to adapting shared information for personal use.

1.1 Methodology

During the summer and fall of 1990, we conducted a set of interviews with thirteen people in seven work groups in a large computer manufacturing company. This was followed by
another set of interviews in the winter of 1991 with fifteen people in one work domain in the same company. Through the interviews, we tried to understand job activities and both successful and unsuccessful information usage from each individual’s point of view.

We did not do a detailed case study in any of the individual work domains. Rather, the focus of our study was on identifying and analyzing the information-related problems and issues that were common to different domains. All of the information issues discussed in this paper were raised in multiple work settings, compensating for the lack of depth in any one domain. We did not prioritize the problems we saw or ask our informants to rank problems.

Most of the people we interviewed had technical jobs; a few did not. Only four of our informants were managers. Though managers are certainly information workers, we chose in this study to concentrate on technical contributors. The work areas for the first set of interviews included technical support, marketing, software integration, finance, electronics product design, and a chemical analysis lab.¹

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 knowledgeable in a wide array of products and located in different countries. They consulted with their customers over the telephone. Their online data sources were combined in a knowledge database which was accessible to all support engineers. This database was constantly updated with previous call histories, known bugs, product newsletters, and other relevant material. In addition, they 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 colocated with their customers and often responded to phone calls or electronic mail requests by physically 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 online 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. They 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.

¹In this last case, we interviewed providers of instruments and software to the lab, rather than the lab members themselves. Our informants had done extensive interviews and observations of their user community.
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.

The work area of the second set of interviews was the design and manufacture of printer products. This included multiple groups cooperating in the design and subsequent manufacture. Outside vendors provided specialized services, sometimes in close collaboration with internal design groups. Information sources here were very similar to those in the measurement equipment design group.

Each interview was loosely structured into three parts. We asked each informant to characterize his or her group and the work processes each followed within the group's overall work process. We then asked the informants to describe the information sources they used (or would like to use) to get their work done. Finally, they were asked to describe specific or general problems they encountered with either the current process or the information sources. In practice, each interview was a conversation that ranged over a variety of topics, some of which we had not anticipated in our preparation. Using a personal, conversational approach can sometimes limit the scope of the data, but it also brings alive the informants' processes and problems in a vivid and thought-provoking way as informants share anecdotes, war stories, and descriptions of informal practices.

In addition to interviews, we were able to observe some of the technical support engineers handling customer calls. During most of our interviews, we were also given tours of the physical setting, online systems, paper filing systems, and other information sources.

2 Individuals' Uses of Information

In the domains we studied, there was often too large a gap between the context in which a worker operated when using information and both the retrieval mechanisms and structure provided for the information. An example we encountered in the printer product development setting illustrates the difficulty of this problem. 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 the task of redesigning 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. He then culled from this stack of material everything related to paper feed design.
Changes are needed in both information technology and organizational processes to support complex, information-rich tasks like the paper feed information search. Assuming that all information about designs had been in an online repository and that all the electronic mail traffic during the design phase had been archived, how could an information management system have helped the engineer do this task significantly faster? Since the data came from varied sources in a very large information domain, the engineer would still have needed help identifying and locating the right designs in the repository. Once the designs had been found, the engineer would have needed to apply filtering and restructuring techniques to the information to help him focus on the parts relevant to paper feeds. Critical information for the task might still have been missing unless some of the design rationale, which is not currently recorded by design engineers, had also been available. This would imply the creation of new reward structures to encourage engineers to write decisions down and the implementation of process changes to allow them time to write extra documentation.

In the remainder of this section, we explore two aspects of the gap between users' task contexts and their information sources. The first area covers the problems of using multiple sources, and the second covers the creation, use, and reuse of custom information compounds. In each area, we describe the existing practices we saw and the implications these practices have for technological and organizational change.

2.1 Using Multiple Sources

Problems in dealing with a single information source are far from solved. Yet the availability of diverse data sources through networked computer systems adds even more complexity to information access tools.

When information workers deal with multiple information sources, they have three problems: identifying the relevant sources among the potentially hundreds of accessible ones, selecting among the relevant sources once they are located, and then mastering the mechanics of interacting with the chosen sources. Interaction is an issue at both the database and user interface layers, since different sources typically require different search mechanisms and offer different capabilities. Note that all three activities apply not only to online sources but also to paper-based and human sources. In this section we discuss each of these problem areas.

2.1.1 Identifying Sources

Before even beginning a search for information, members of the marketing group needed to decide which of their fifty electronic mail folders, corporate databases, and personal files to include in the search. Some of the electronics product designers had physical access to hundreds of file clusters created by other designers at different locations and containing potentially useful designs, but the engineers we interviewed rarely viewed the online designs of other engineers. Exceptions to this rule occurred in a small group of engineers working on related parts; they occasionally viewed each other’s designs to make sure that the interfaces between parts looked correct. In a rich information domain, it is very difficult to identify the appropriate and available sources for a task at hand without prior knowledge of the sources’ existence and applicability.

As computing environments become more completely interconnected, identifying proper sources is difficult enough to be an issue for information system designers. The tools operating in such environments must be able to display road maps of available sources and must
include a top-level description of the sources’ physical limitations and scope. Tools should flag access path problems and permission-based limitations. The user should not even be shown clearly irrelevant sources and should be made aware of existing but currently inaccessible sources. Ideally, the user should be able to provide more information to the tools so that more refined choices can be made about source relevance within particular contexts. As the number of sources grows, techniques for locating specific sources within a large set of choices should be just as effective as those used to locate information within individual sources.

One difficulty we found in this context was that there was often no organizational or technical conduit through which the existence of a source could be made public. Often, engineers suspected that sister organizations might have produced a mechanical or electronic design similar to the one they needed, but there was no forum for publishing such information, even though the designs themselves would have been accessible through an electronic network.

2.1.2 Selecting Sources

Once a set of potentially relevant, available sources has been located, an often very subjective process of selection takes place. A first impulse would be to help a user avoid this step altogether by integrating sources to such a degree that they appear as a single repository, thus avoiding the selection step altogether. We found, however, that such source transparency would destroy important information: users in several groups wanted to know where the various pieces of information were coming from.

For example, in the technical support group, some engineers judged the strengths and weaknesses of overlapping records of previous problems based on who had created the record, when it had been created, validation policies for information entry, and the format used to describe the information. They were able to use that knowledge to improve the quality of information they extracted, by choosing to search a limited set of data sources (such as previous customer calls, certified entries, or known bugs) or filtering entries based on attributes such as the author’s name.

Individual style differences affected source selection in some settings. For example, in the marketing group, some people gathered competitive analysis data mostly by attending trade shows and collecting data sheets to read. Others preferred to use their contact networks, collecting information mostly through conversations. In this domain, there was tolerance for differences and recognition that the results mattered more than the technique. Individual differences are discussed further in Section 3.2.

The age of a source’s contents and the lifetime of information in its domain also played a part in the selection process. Marketing information, for instance, tended to have a lifetime of about six months, much shorter than the lifetime of information on designs or existing products. Sources that were infrequently updated were therefore useless to some workers but quite acceptable to others.

Selection preferences sometimes included the typical progression from one source to another. One technical support engineer started each search with certified documents, which had been reviewed before being entered, then progressed through other online sources until information from a human being was the only possibility left. To consult people, the engineer then proceeded, by increasing distance of the work group, from friendly coworkers to organiza-
tionally neighboring groups or contact persons in the operating divisions that produced the relevant product.

Such a preferred progression through sources should be supported by information tools. Other factors on which tools should be able to give information include access costs, performance, usage frequencies, and backup policies for various sources. In addition, a user should be able to convey to the tool parameters regarding confidence levels or other selection criteria.

These uses of knowledge about sources to support selection imply that many of the distinctions between the sources should be maintained. But once a set of sources is accessible through a common interface and access speed differences are not too great, some of the distinctions between different sources may no longer be perceived as particularly significant, though different sources will still be more or less valuable for different uses. The source of a piece of information should become another one of its attributes, used to help people find, recognize, and choose information.

2.1.3 Interacting with Multiple Sources

A great many information workers do not yet have the advantages of integrated source interaction. They manipulate a variety of sources, each in accordance with its own interaction mechanisms, and they try to extract coherent information from the aggregation. The lack of integration is sometimes due to missing support, but many times the sources are just too different to integrate successfully. Interestingly, workers from several different domains used the same phrase, “one-stop shopping”, as a metaphor to describe the way they wanted their interactions with varied information sources to feel. Though actual merging of sources is not an appropriate or practical goal, one-stop shopping implies providing easy access to multiple sources, with common mechanisms for retrieval and manipulation of data.

Sometimes different data sources used in the same task have little or no semantic overlap. For example, a marketing manager might consult both structured price lists in a corporate database and free-form customer questionnaires in order to make pricing decisions. Here, the user wants to interact with both sources at once, perhaps comparing data from each and clipping out subsections to include in a pricing proposal for a new product. These actions require an infrastructure that permits some joining of autonomous sources.

Even when information sources do have semantic overlap, there may be practical constraints that make the cost of source integration too high. Where there is a large investment in externally produced, unintegrated software tools, as in product development, integration may not be cost-effective. The data produced with the tools is extremely valuable and the product groups are under great time pressure, so any changes to the tools must be minimally disruptive and maximally cost-effective.

In other cases, the disadvantages of unintegrated sources are not considered to be very great. In a chemical analysis lab, for example, the data sources are often instruments. Some data will probably never flow smoothly from one instrument to another, but will always be transferred by manual entry. The weight of a sample is an example of this kind of data; scales are separate instruments and likely to remain that way.

Attempts are being made to provide integration for sources whose contents overlap (see [3, 4, 5, 6, 7]), but many problems remain, particularly with reconciling differences in the information contents. We can distinguish between structural and semantic heterogene-
ity among information from various sources [8]. A structural difference is a variation in the representation of an information item. An extreme example in product development is the multiple descriptions used for new mechanical designs. One description of a new assembly is a list of textual part descriptions with additional information about their placement and interconnections, which is used to evaluate manufacturability and influence purchasing schedules. Another description represents much of the same information as a 3-D model, created by the engineer as part of the design process. When representations do not fit, as in this example of mismatched file-naming conventions described by a design engineer, painful manual record-keeping sometimes takes over:

People bought all these CAD tools, but all of a sudden they realized that they were swamped with tons and tons of files laying around and no way to take care of them... Usually, each tool had embedded in it some capabilities to access the file system and do a little bit of file management, okay, but they were very tool-centric and each tool did it a little bit differently, so you still had a lot of people keeping lists of files on the backs of envelopes stuffed in drawers.

Semantic heterogeneity consists of information that is present in one but not the other source or differing vocabulary usage among sources. An example of differing contents in two design databases is the inclusion of a mechanical part's tolerances in one source when such information is not maintained in the other. An example of differing vocabulary usage is differing orientations in numbering the pins of an integrated circuit chip. These potentially extreme differences among sources make general solutions to seamless source integration very difficult to achieve, although tools can make progress in individual cases.

Even if diverse sources could be integrated cleanly, there is the problem of different workers having different preferred interaction modes. Individuals have preferences in metaphors and terminology, as well as differences in training and background. Someone who is very adept at interacting with a particular marketing database might well refuse to change to another interaction style just because the new style unified interactions with other sources.

In spite of these difficulties, it is important in most cases for information tool builders to attempt integration. People find it frustrating to reenter the same information into two different access tools or to use two different search methods because the sources are not integrated. We have seen users having to change to different terminals to access different sources. Having the same incompatibilities in a workstation environment between two windows next to each other makes the limitation even more painfully clear to users. One engineer described his need for tool integration this way:

Having tools that come from a variety of disciplines that work together well could provide a lot of flexibility... if you could use your favorite word processor to do annotations... if people found it easy to extract numerical information out of their designs and could be used in spreadsheets easily...

Given the investment in tools and data sets, often accumulated over years, we must expect integration problems to be a hindrance for a number of years to come. It is important to remember that integration need not be complete or seamless to be successful.
2.2 Custom Information Compounds

Though the uses of information varied widely among the groups we studied, one element these groups had in common was the need to unite related pieces of information in one collection that either helped solve a particular problem or could be used as an end product for a task. We call these collections information compounds.\(^2\) The information compounds we saw were often used repeatedly, although sometimes they were abandoned because they had only temporary value. In all systems we saw, compound building was inadequately supported.

Fully developed information compounds would help users make task- and role-based selections of information from large data stores. Compounds would provide a powerful filtering function, by limiting the information that was retrieved and presenting it in a meaningful way. An example of a system that tries to support some compound building as part of data source browsing activities is described in [10].

In 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 in many domains.

2.2.1 Building Compounds

Information compounds are potentially very useful, but they are not easy 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 that well from a couple of different angles. You know, I doubt if you'll find the perfect one 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 be supported? There are two ways to leave a record of a problem: 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

\(^2\)In the context of hypertext nodes, compounds are similar to the "composites" outlined by Halasz [9].
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 other 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 the creation of a separate entry for each component of the solution and a referential compounding solution to tie them together, 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 3.2.

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, online world. Important information might be available only in an inaccessible medium or a temporarily unreachable data source.

Apart from access or 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.

When compounds are stored for future reference, maintenance of the archive is 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. Whether to create a static or dynamic compound should be left to the discretion of the information handler, without overloading him or her.

2.2.2 Capturing Filtering and Retrieval Expertise

Filtering reduces the effective size of the information space and should make it easier to deal with. When the amount of information is very large, a filtering capability is a necessity. Information compounds are a generalization of filters, in which the information may come from multiple sources rather than a single source. Retrieval and presentation expertise play an important part in the creation of information compounds and other filtered views.

Often the information tools in the domains we studied provided filtering actions as one-time requests. However, many filtering requests comprised expertise which had the potential to be reused, which implies that retrieval tools should allow useful filters to be saved. This includes expertise in identifying which information is relevant to some given task as well as expertise in retrieving it.

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. As soon as such patterns are perceived, users should be able to have the system save the expertise captured in the pattern so it 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 online systems the engineer used to do much of his work, but nevertheless sufficiently valuable that he took the time to create and maintain it.
Capture of retrieval expertise is especially important because expertise is an asset which is not easily found and transferred throughout an organization. People who know their way around an information system often become highly valued local experts; we saw this in the internal and external technical support, software integration and marketing groups. Some of the experts' knowledge in the form of their retrieval patterns can be shared.

In the software integration and technical support groups, we did see some people writing notes or reports describing what they knew about domain-specific information retrieval tasks. However, these efforts were not considered completely successful by the groups with whom they were shared. A primary difficulty seemed to be in organizing the notes to support a variety of retrieval tasks, some of which were unanticipated by the authors of the notes.

Reusing queries affords an opportunity for an element of personalization in information systems. We found that individual engineers had differing opinions regarding the importance of the age of a piece of information and the reliability of particular contributors. Retrieval mechanisms can capture such predispositions in permanent filters valid for a particular user. For a discussion on more sophisticated retrieval expertise, see [11].

2.2.3 Structuring Information for Multiple Uses

When information providers can predict all uses of information, an optimal data model and organization can be chosen for storing the data so that querying, filtering, and browsing are straightforward and well matched to the users' tasks. Unfortunately, this is seldom the case. The difficulty of structuring information for multiple uses contributes to the gap between information systems and users and 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 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 marketing was shared both with research and development groups and with upper management in different formats and to different ends; 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.
Task needs primarily affect the structure of information as it is evidenced in retrieval mechanisms, but retrieval needs often influence storage mechanisms as well.

When the structure of information is fixed for all users, it can seem mismatched to the information's use because of personality differences among information workers. As was documented in [12], people have differing needs for organization in their 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.

In domains where information is likely to be used in multiple, unanticipated ways by users of different temperaments, developers of information systems should find it a useful strategy to provide 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 appeared. One way to do this is through the creation of information compounds for different users, roles, and tasks. Compounds may be specified directly by their users, or they may be created by information access experts in consultation with users.

3 Information for Group Use

We now turn to the second cluster of information-related issues: how shared information was used by work groups. First, we note that some people record information strictly for personal use, to help keep track of multiple tasks or hard-to-remember details. Activity logs, Post-its, to-do lists, and private notebooks are examples of personal records; we saw many such examples in the domains we studied. It seemed that part of the character of each individual office was expressed in how the worker chose to arrange personal information artifacts — spread on desks, posted on walls, or attached to workstation screens.

However, a great deal of information was recorded in order to be used by someone other than the creator. Even lab notebooks have dual individual and group use; they are invaluable as personal design records, but they also play an important role both as resources for engineers who are downstream in the production process and as trails needed for declaring patents.

Information meant for group use is different from information for personal use. There are often barriers to information flow and information understanding, depending on the reasons for sharing and the communities involved. To understand some of the complexities of shared information, let us return to the example of 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, then
called on the designers who were still with the company to guide him through the material
and provide additional context. The use of intermediaries can be a very successful strategy
for dealing with shared information resources.

There are many more examples of shared information in product development. In a design
team working together closely on a product, each participant must coordinate his or her
efforts with coworkers. When changes are made, team members must be notified and new
dependencies must be recorded. Manufacturing and test processes that are being developed
in parallel may also be affected; different professionals need different views on the growing
collection of data. To help engineers make cost-effective decisions, company-wide parts and
vendor databases must be made available. In some cases, however, these databases may not
contain the specific information about parts needed by a particular group or a group may
have more up-to-date information on new parts, so group members may develop local or
even individual versions that supersede the more distant information sources.

Greif and Sarin [13] discuss the technological support needed for data sharing in 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 infras­
tructure 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, the situation is more amorphous. There are certainly examples of
within-group sharing of well-defined data, but in addition, cross-functional and cross-group
sharing are common. Information is often ill-structured. The same information source may
be used by individuals who have different expertise, different tasks, and different ideas of
what the important information objects are.

In this section, we discuss five aspects of information for group use that appeared in the
complex information worlds of our domains: barriers to sharing within different communities,
personalization of shared information, common characteristics of shared information, the use
of intermediaries or information brokers, and maintenance of shared information.

3.1 Sharing Within Circles of Trust

In the groups we studied, sharing of information most readily occurred within a small com­
munity 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. The 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.

Within a group, there are also assumptions of shared background and goals in using the
information. When information is disseminated close to its source, the recipient is likely to
be able to use it without much need for generalization or explanation. When information is
exported to a diverse group, the producer must often supply additional context in the shared
information, or the reader must infer such context. Diversity can come from different levels
of expertise within a common discipline, as well as from differences in disciplines or roles, as a member of the marketing group pointed out:

We spend a lot of time trying to figure out ways to represent it 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.”

In a large, homogeneous group such as technical support engineers, the common job culture and tasks led to fairly effective sharing of information. The database of previously-solved problems was considered successful, although its entries were created by hundreds of different people. This attitude was aided by the fact that in this culture, engineers were encouraged to develop broad expertise rather than narrow specialties and to use certain common problem-solving approaches.

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. Marketing engineers often shared potentially unreliable, incomplete information; it seemed to be part of the culture of that group. But in technical support, histories of customer calls were not generally accessible 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, which inhibited the problem-solving process. We did see 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.

The dissemination of incomplete, preliminary information is dependent on trust. Tools operating in settings where such early dissemination is an issue should allow users to specify “circles of trust” over the user community and should allow information to carry reliability and stability attributes.

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 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, managers may not want the people they manage to advertise their expertise, since that might bring them an undesirable workload.

Terminology can also be a problem. Within a group, terminology is usually well understood, but it is still far from standard or unique. Even using different type fonts for the same terms can trip up some search mechanisms. 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.

Thesauri and related mechanisms are currently being used, mostly to facilitate knowledge-based retrieval by enlarging term sets. We suggest extending this usage to use thesauri to support sharing and communication within groups. For more discussion of thesauri in information retrieval, see [14, 15].

### 3.2 Personalization of Information

Where there are large, shared information spaces such as the knowledge database in the technical support group, a crucial problem is to provide ways to personalize or import the information into an individual’s space and tailor it to his or her needs. People’s levels of expertise and individual working styles may be vastly different, affecting their patterns of information production and use. The effectiveness of information created for group use is measured as much by the consumer’s ability to personalize it sufficiently as by the producer’s ability to disseminate it successfully.

#### 3.2.1 Detectives and Librarians

We found marked differences in the approaches and attitudes of 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. One manager had written a scenario describing extremely different problem-solving styles that was widely distributed among the support teams; some of its descriptive labels had become part of the common job language.

One of the engineers we interviewed 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 online 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 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. Many problems, of course, can be solved using either approach. 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.

Greif and Sarin [13] discuss the use of roles to describe collections of access rights that could be assigned to individual users in a group, as a way of coordinating work on shared data. The approach is described as “abstracting and packaging defined patterns of behavior.” Although in Greif and Sarin's work roles refer only to access rights, they might be generalized to interaction styles. This role characterization expands on the traditional user model, which attempts to describe the kind of information the user wants to access, and attempts as well to describe how the interaction should feel.

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. If the system were to insist exclusively on such fine categorization, acceptance among the detective population would suffer and the system as a whole could be adversely affected.

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 work could alleviate the tedium of copying information from a work area into a call history report. A simple feedback mechanism could summarize 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.

3.2.2 Formal and Informal Roles

Work roles and processes change as new technology is introduced, and they can influence technology needs in turn. Understanding the interactions between organizational and technological change is difficult, but it is often necessary if changes are to be successful. An internal tool developer in product development argued eloquently for taking time to think about these issues:
You have to understand the process before you add tools, because tools have a tendency to screw up the process, if you didn’t know what it was in the first place, now you have another one you didn’t know what it is and you thought the tool was going to save you and in fact, it didn’t. So understanding the process is absolutely essential, it’s the root, it’s the central core, if you will, of getting things done... If you can’t identify what you’re doing every step of the way and you don’t recognize what the process is doing and you change the tools within it, you could destroy everything you had working that was good.

To begin with, different role assignments create different needs for information technology. 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.

Sometimes people choose to take on informal roles within their work groups. When new technologies enter a work setting, people develop new skills as they learn how to put these technologies to use. Often local experts emerge [16]; these are people who understand particular tools especially well and can give advice on their use. The acceptance of tools can depend on such champions, and their activities must not be seen as extraneous by the organization. If organizations are slow to reward the contributions of the champions, the use of new technologies may be effectively discouraged.

We saw an example of this in the product development groups. To improve the design process, electrical engineers were given sophisticated simulation tools. To use simulations well, it is necessary to spend time with the tools in order to develop expertise. This requires both a process change (allowing more time for simulations) and a recognition on the part of the organization that acquiring the necessary skills is worthwhile. Some engineers felt that the highest rewards were reserved for generalists and for people whose direct contribution to a product was obvious, and that it was not in fact in their best interests to learn this special area in depth or devote time to its practice. Hence, they felt that simulation techniques were underused in their work settings. The internal tool provider put it this way:

If some person is sitting over here just managing data, what is their added value to the product? ...You will have engineers supporting other engineers that are not creating product... If you don’t have that enabling team, it just won’t get done, it just doesn’t get done.
Just as information technology can influence the work force, a changing work force population might require system designers to respond with new designs. For example, analytical chemistry labs are finding it increasingly difficult to find new chemists; the supply does not meet the demand. Labs have begun to use more technicians, whose level of education is lower and length of stay with the organization is often shorter. Tool providers in this setting have been required to pay much closer attention to ease of use and consistency across applications.

3.3 Interwoven Information

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 are not necessarily defined as a group of people who work for the same manager, though often this is 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.

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.

The information people shared with outsiders tended to be simpler, with less context. 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.

In the domains of our study, much of the shared information related to work interdependencies, among both insiders and outsiders. Malone and Crowston [17] outline a theory of coordination, defining coordination as "the act of managing interdependencies between activities performed to achieve a goal." They give a preliminary list of the different kinds of interdependence and the common information involved. We suggest that the same interdependence relations they discuss (e.g., prerequisite or shared resource) may be expressed through very different kinds of information, based on the closeness of the community in which the interdependence occurs.

In this section, we will consider examples of both insider and outsider information, with particular emphasis on the complex, interwoven information shared with insiders. There was a large variety of interwoven information in most of the domains we studied, and it was considered both extremely important and difficult to express and manage by many of the people we interviewed.

Let us look at a detailed example of interwoven information in the technical support group. The call histories created by the support engineers were used for a variety of purposes. A call history was a record of a problem-solving activity that consisted of a time-based flow of events, intermixed with notes showing the engineer's growing understanding of the problem.
It contained accumulated context, false trails, and backtracking. To learn new insights about problem-solving, the user would want to follow all of this. To simply reuse a solution, the user needed sufficient context to decide whether the call was relevant to a current problem, but then needed to be able to go straight to a complete solution description. An audit trail needed a pure chronology of events, with time-stamps. Finally, the use of a call history for providing feedback to internal product developers required a higher level of analysis to infer a design or implementation weakness from the customer's problem.

In the technical support example, a call history described a particular execution of a process. A process description could also be a template describing a prototypical process. Consider the information shared by workers in the software integration group. This group had difficult coordination tasks, both with their developer partners and within their own group, since several integration cycles were conducted in parallel and managed by different group members. Specialized expertise and roles 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 enable them to wear one another's hats. A member of the integration group described a frustrating scenario involving process information:

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 online 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^3 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.

---

^3 "System.group" is a technical jargon term used by the system integration team.
Process-related information like the call histories and software integration handbook 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 should be complete enough to allow users to "tap in" at many different points. In customer support, for example, a new problem might be a more specific or more general case of an old problem; the old problem description would still be of value if search mechanisms through the problem description were sufficiently flexible.

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.

Not all interwoven information consists of process descriptions. Explicit descriptions of interdependencies and other linked data are also common types of complex insider information. For example, in the product development groups, investigation reports, design information, parts and inventory lists, and schedules were shared. This information was used to gather consensus among the diverse people in a product team and keep the group on track. However, design overviews or downstream detailed views were not always updated when changes occurred during design; the design record was not necessarily updated when changes were introduced during the manufacturing stage; master schedules were sometimes less up-to-date than individual schedules. The difficulty of managing this information lay partly in the need to understand and track interdependencies between the various sources, which in turn modeled work interdependencies. Here, the absence of links among interrelated pieces of information was a real barrier to effective sharing.

Information shared with outsiders, by contrast, 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.

Sometimes the ownership of a customer problem had to migrate from one support group to another, if engineers believed that the second group had more expertise in the problem area. 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.

In the product development groups, there was an interesting variation on the insider/outsider notion. In this setting, the boundaries of community were unexpected and sometimes fluid. Parts vendors seemed to be clearly outsiders, yet they were in some cases working in collaboration with manufacturing process engineers inside the company. It would have been very convenient to directly provide vendors with online 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.

3.4 Information Brokers

Both people and computers have their drawbacks as information sources. One of the technical support engineers explained the difference: "Computers are terse and unfriendly; humans are scattered and have ego problems." The "terseness" and "unfriendliness" of computers is especially apparent when online information is shared among diverse groups of people, because of the many barriers to sharing and understanding we have outlined above. People in these situations sometimes choose to use other people as information sources.

People excel as sources for particular kinds of information, especially information that is imprecise or needs expert filtering. Sometimes people act as interpreters to online information, providing explanations not present in the data. We have seen an example of this in the product development groups, where engineers acquired most of their information about old designs from the original designers rather than the documentation these designers left behind.

In some settings, we found information brokers who performed a matching service to help clients find information sources. 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 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 expertise 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 online telephone interface, she might know about buttons in a Motif window system.” In environments with a large number of heterogeneous sources, this kind of expertise in identifying and selecting sources is invaluable and can greatly facilitate the sharing of information.

Let us imagine a simplistic online “expert directory” that performed some of the activities that brokers currently handle. Such a model would probably require experts to make themselves known to the system. The trouble with this requirement is that people are hesitant to take such an official-looking step: anything written, even an entry in a contact database, has the implication of accepting responsibility. Since this is an informal assistance network, such appearance of formal responsibility would be inappropriate. Another reason experts might not sign up is that some people do not consider themselves experts in a particular area but are, without realizing it, the most knowledgeable person available locally. Clearly any online directory solution would also need strict access control mechanisms, and it would not be able to support the sophisticated matching that human brokers do.

These observations reaffirm that an information system designer must be aware of the underlying organizational and social factors of the status quo before introducing automation. In some cases these factors will significantly affect the design. In other cases they can make a good computerized solution impossible.

### 3.5 Source Maintenance

Upkeep is a major problem with information sources. Sources may become outdated if new information is not added or if incorrect information is not revised.

We found the latter to be a difficult and controversial issue. The amorphous information prevalent in most of the domains did not allow easy tracking of interdependent entries, so it was relatively easy to omit necessary updates. Information in data sources was sometimes used as a history of events within the organization, so making changes meant losing part of the historical record.

Another difficulty lies in determining who has the responsibility of creating or updating the information and who would benefit from its being up-to-date. The success of information systems may well depend on the responsiveness of the information providers; it is therefore important to make sure that these people will be motivated to do their part. However, the control and use of diverse sources is often distributed among work groups in different parts of the organization, and some sources may be controlled entirely from outside the organization. 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.

Technical support engineers were themselves encouraged to create new "engineering notes" by the existence of a suggested monthly quota. These engineering notes were carefully prepared statements of a problem and the associated solution. However, there was no particular encouragement to make corrections or additions to engineering notes. Once completed, they were never changed. If maintenance activities are to take place, the organization must somehow reward them.
4 Summary

The following tables summarize the problems and issues information tool designers should learn about the domain and user community they are targeting. Each table lists domain facts to learn, issues to pay attention to and some relevant literature. Clearly, designers must be prepared to discover other factors and forces in their domains, but we offer the following tables as a guide:

<table>
<thead>
<tr>
<th>Information Sources</th>
<th>Domain Facts to Learn</th>
<th>Related Factors</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>What are the current online sources?</td>
<td>Media, location, limitations, lifetime, interoperability, reliability, accessibility, security.</td>
<td>[18, 3]</td>
</tr>
<tr>
<td></td>
<td>Which information comes from people – why?</td>
<td>Brokers, specialists, subtle interaction conventions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Which information comes from paper – why?</td>
<td>Legal restrictions, convenience, psychological reasons, mobility.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>How are transitions between different sources managed?</td>
<td>Formal and informal guidelines, tools.</td>
<td></td>
</tr>
</tbody>
</table>
### Information Contents

<table>
<thead>
<tr>
<th>Domain Facts to Learn</th>
<th>Related Factors</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>How is information structured?</td>
<td>Free text, regular/irregular, check user's internal models.</td>
<td>[19, 20]</td>
</tr>
<tr>
<td>Is information structure changing?</td>
<td>Who needs to change it? Under what circumstances?</td>
<td></td>
</tr>
<tr>
<td>Is non-factual information stored and used?</td>
<td>Process, histories, deltas, constraints, context.</td>
<td></td>
</tr>
</tbody>
</table>

### Information Usage

<table>
<thead>
<tr>
<th>Domain Facts to Learn</th>
<th>Related Factors</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are compounds used?</td>
<td>Static or live, to be saved or disposable.</td>
<td>[9, 10]</td>
</tr>
<tr>
<td>Are ad-hoc or imprecise queries used?</td>
<td>Look for retrieval patterns.</td>
<td>[21, 22, 23, 24]</td>
</tr>
<tr>
<td>Is retrieval expertise identifiable?</td>
<td>Find specialists, macros, softkeys or scripts being used.</td>
<td>[11]</td>
</tr>
<tr>
<td>Is browsing done?</td>
<td>Large search spaces? Can structure be used for support?</td>
<td>[25, 26, 27, 28, 29]</td>
</tr>
<tr>
<td>Are multiple views and terminology used?</td>
<td>Check professional roles of multiple users, cross-check usage in multiple contexts.</td>
<td>[30]</td>
</tr>
<tr>
<td>Is information archived or moving?</td>
<td>How do users cope with monitoring information?</td>
<td></td>
</tr>
<tr>
<td>Should sources be active?</td>
<td>Information ‘falling through the cracks’, need for coordination of work, data consistency problems.</td>
<td>[31]</td>
</tr>
<tr>
<td>Is information shared?</td>
<td>Understand circles of trust.</td>
<td>[14, 15, 30]</td>
</tr>
<tr>
<td>What are current barriers to sharing?</td>
<td>Diverse population, practical obstacles, approval requirements, user resistance.</td>
<td></td>
</tr>
<tr>
<td>Are heterogeneous sources in use?</td>
<td>Identification/selection/interaction mechanisms.</td>
<td></td>
</tr>
</tbody>
</table>
### Information Maintenance

<table>
<thead>
<tr>
<th>Domain Facts to Learn</th>
<th>Related Factors</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is information deleted?</td>
<td>Legal restrictions, resource restrictions, psychological factors.</td>
<td></td>
</tr>
<tr>
<td>Is information modified?</td>
<td>Legal restrictions, dependency maintenance, motivational issues.</td>
<td>[31]</td>
</tr>
</tbody>
</table>

### Organizational Issues

<table>
<thead>
<tr>
<th>Domain Facts to Learn</th>
<th>Related Factors</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ‘official’ work process</td>
<td>Examine different people’s viewpoints. Is work event-driven or planned? Are people frequently interrupted?</td>
<td></td>
</tr>
<tr>
<td>The actual work process. Deviations from the official process.</td>
<td>Coordination and interaction mechanisms. Shortcuts, tolerated exceptions.</td>
<td>[2, 32]</td>
</tr>
<tr>
<td>Reasons for deviations?</td>
<td>Official process flaws, temperamental differences, power plays.</td>
<td></td>
</tr>
<tr>
<td>Process breakdowns.</td>
<td>Crises, missing information, missed deadlines. Is the process fundamentally flawed or just poorly supported?</td>
<td></td>
</tr>
<tr>
<td>Skill distribution.</td>
<td>How might an information system change the distribution?</td>
<td>[33]</td>
</tr>
<tr>
<td>Individual differences in temperament and work habits.</td>
<td>Speed, search techniques, communication patterns. Librarians/detectives.</td>
<td>[34, 35]</td>
</tr>
<tr>
<td>Formal and informal specialist roles and how they are used.</td>
<td>Organizers, tool usage innovators. How and why workers access the services of specialists.</td>
<td></td>
</tr>
<tr>
<td>Turnover rates.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Characteristics of novices and experts.</td>
<td>Learning times, particular learning problems, teaching techniques used.</td>
<td></td>
</tr>
<tr>
<td>Reward system.</td>
<td>What do people find satisfying? What is valued/not respected?</td>
<td></td>
</tr>
<tr>
<td>Metrics.</td>
<td>How are metrics used in the work process? Do people feel threatened by them? How could they be useful?</td>
<td></td>
</tr>
</tbody>
</table>
5 Conclusion

Our interviews brought out some of the deficiencies in the current level of technical and organizational support for information handling. To address these deficiencies, it is essential to understand the real-world complexity of managing information in the technical workplace. This complexity is revealed in the variety of interrelated information sources, information-intensive tasks, and scenarios of information use.

Information sources can be online or offline, 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 often has 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 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.

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. In this paper, we have presented an initial description of the technical and organizational changes that are needed to support information workers.
6 Acknowledgments

The essence of this paper is the information related to us by the people we interviewed. In spite of their workloads, they spent hours demonstrating their activities and sharing their thoughts. 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. Finally, Bill Cohen provided great help with the background literature search.
7 References


