Published in the Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98), Seattle, WA, November 14 – 18, 1998.

Just Talk to Me: A Field Study of Expertise Location

David W. McDonald and Mark S. Ackerman
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
{dmcdonal, ackerman}@ics.uci.edu
http://www.ics.uci.edu/CORPS/dmcdonal
http://www.ics.uci.edu/CORPS/ackerman

ABSTRACT
Everyday, people in organizations must solve their problems to get their work accomplished. To do so, they often must find others with knowledge and information. Systems that assist users with finding such expertise are increasingly interesting to organizations and scientific communities. But, as we begin to design and construct such systems, it is important to determine *what* we are attempting to augment. Accordingly, we conducted a five-month field study of a medium-sized software firm. We found the participants use complex, iterative behaviors to minimize the number of possible expertise sources, while at the same time, provide a high possibility of garnering the necessary expertise. We briefly consider the design implications of the identification, selection, and escalation behaviors found during our field study.

Keywords
Expertise networks, knowledge networks, computer-mediated communications, expert locators, expertise location, expertise finding, information seeking, CSCW, computer-supported cooperative work.

INTRODUCTION
Wayne is a support rep for a software company. Today, a client sent Wayne a problem, one he hadn't seen before. Wayne spent several hours trying to diagnose the problem and eventually realized that he could not solve it alone. Now Wayne has another problem — deciding whom he will ask to help him. Unfortunately, this is not simple; Wayne is not sure of the problem's exact cause. Is the problem related to user training? Or, is the problem the result of a recent customization by the developers? Will Wayne need to resort to just asking around?

Wayne must find the expertise needed to solve his client's problem. He needs to find the people who are most appropriate to ask and get their help.

Systems that assist users with finding others who have specific, desired information are increasingly interesting to organizations and scientific communities. However, as we begin to construct such systems, it would be useful to determine the social, cognitive, and informational aspects we are trying to augment before constructing these systems. It is our goal to guide the construction of information systems that facilitate finding expertise within organizations. We believe that one can do so best with a field study of existing practice. Accordingly, we conducted a five-month field study, using a socially distributed cognition viewpoint, of how people in a medium-sized software firm find the expertise to construct, maintain, and support their software systems.

We begin with an overview of the literature covering expert-locator systems and several key studies of the collaborative aspects of expertise. We then detail the field site, the data, and the site's work and information needs. Following, we describe the site's mechanisms for identifying and selecting people with required expertise, as well as show how the organization prevents or recovers from breakdowns. Finally, we conclude with implications for future systems.

CSCW AND COLLABORATIVE INFORMATION SEEKING
The CSCW and adjacent literatures have numerous attempts to facilitate finding information through social networks. Several systems have explicit models of expertise.[1] The Answer Garden [2] and Answer Garden 2 [3] systems, designed to facilitate informal flows of information and their capture, have a naive view of expertise. Answer Garden allowed only a few levels of expertise in answering questions for users. While Answer Garden 2 had an explicit expertise-location engine and provided computer-mediated communications mechanisms to find others with a range of expertise, the mechanisms were not very elaborated.

Two other systems help find people knowledgeable about a topic. Referral Web [12] helps find research experts. Refer-

[1]In this paper, the term *expertise* assumes the embodiment of knowledge and skills within individuals. Our definition distinguishes expertise, which is a range, from expert. An individual may have different levels of expertise about different topics. Expertise can be topical or procedural and is arranged and valued within social and institutional settings.

ral Web assumes a topical expertise among all co-authors, and therefore identifies expert individuals by their participation in co-author relationships. Yenta [7] creates and shares personal profiles based on text content analysis. It can then route messages along a network of inferred common interest (based on keyword or other indexing techniques). Both of these systems not only assume a single level of expertise, but also that one can determine a list of experts through merely textual analysis.

While collaborative filtering systems [9, 13] can help find other people, they generally do not distinguish levels of expertise. A related system, Phoaks [10], can be used to identify who contributed information. This distinction among contributors allows a user to make judgements about the contributors' expertise, but these must be inferred.

None of these systems have very sophisticated models of expertise and information seeking. The social world is far more complex. Studies of collaborative problem-solving make this clear. For example, Allen [4] noted that engineers differentiate whom to ask with great nuance. Information seekers weigh the psychological cost of asking, including loss of status, expected reciprocity (i.e., likelihood of returning the favor), and social equity (i.e., how well they know the person socially).

Cicourel's study [5] of medical diagnosis provides ethnographic detail about the social complexity surrounding information seeking in organizations. Cicourel demonstrates that the expert's knowledge, organizational status, and day-to-day interactions contribute to the attribution of expertise and authority by those who work with the expert. This complex set of reinforcing mechanisms supports and enhances the status of an expert as an expert. Additionally, Cicourel noted that both information seekers and experts make judgements about the validity of the information they receive. The specific structure of the organization in Cicourel's study (i.e., a teaching hospital) led him to examine the structured interchange of expertise within daily face-to-face meetings.

Orr's study [14] of copy repair technicians provides additional ethnographic detail about the social complexity of expertise. In the technicians' work, learning by doing and through stories are the main mechanisms for displaying, sharing, and propagating expertise. Idiosyncrasies of the customer and their equipment, incomplete information about the state of a breakdown, and the relatively independent nature of the technician lead to expertise that cannot be formally codified or categorized. Like Cicourel's doctors, the service technicians in Orr carefully weigh the selection and quality of information sources. As well, their use of their expertise in a diagnostic situation is constrained by many organizational and social factors. However, Orr's major concern is the use of narrative in diagnosis and information sharing; he is only peripherally concerned with seeking expertise per se.

Within this study, we extend Allen's work in identifying potential sources of information. We have adopted Cicourel's and Orr's general point of view to examine finding expertise in technical support, bug tracking, and similar software activities. These work processes occupy a middle ground between the life-critical and highly personally interdependent medical work in Cicourel and between the relatively independently accomplished service work in Orr. We believe that the type of work processes we examined are common within workplaces, and our emphasis is on locating expertise within daily work.

Next we turn to a description of the study and site.

THE STUDY SITE AND DATA

The study took place at Medical Software Corporation[2] (MSC), a company that builds, sells and supports medical and dental practice management software. MSC sells its software pre-installed on a high performance Unix server as a turnkey system. MSC has been in this business for almost 20 years and they have a large share of the group medical and group dental markets. MSC is a medium-sized company, with just over 100 employees.

Practice management software provides functions distinct from clinical management software. Practice management is primarily concerned with appointment scheduling, treatment planning, patient recalls, insurance billing, patient billing, and payment reconciliation. These functions are often considered the business side of medicine. These functions are closely related to the clinical management (treatment) of the patient.

The majority of the participants in the study worked in three departments, Technical Development, Technical Support, and Technical Communications. Each department provided access to entry-level, senior, and management employees. Additionally, we observed and interviewed several people in Human Resources and Client Relations.

The first author was on-site for 5 months and conducted follow-up interviews for another 4 months. Data were collected through participant observation, semi-structured interviews, and informal open-ended interviews. In total, we conducted 37 formal interviews and more than 50 informal interviews; formal interviews were taped and transcribed. Additionally, we had access to various internal communications including memos, internal and client oriented electronic bulletin board systems, and a large number of online historical files.

We analyzed the data using standard ethnographic techniques [17]. We approached the data using Socially Distributed Cognition (s-dcog) theory [11, 16], since it considers the expert within the social context of the information seeker. S-dcog provides a useful view of the cognitive as well as the social aspects of the organization and its knowledge resources. Following the common s-dcog framework,

[2]All names and identifiers have been changed in this paper.

the socially cognitive aspects are observable through processes that translate cognitive states among levels of representation or where representations are effectively externalized in some form. Viewing asking for help as an externalized representation that becomes shared among multiple participants through time afforded a fresh perspective on information seeking activity.

THE NATURE OF THEIR PROBLEMS

The types of problems which MSC employees face stem from the complexity of the software and the expansive feature set. An appreciation of the complexity that these informants face on a day-to-day basis is important to understanding why they work so hard to find just the right person to answer their question or help with a problem. The medical system is a good example of the complexity of MSC's systems.

The medical system has a text based interface and is often used with text only terminals. The top level menu of the medical program has over 100 specifically identifiable features, and it is common for there to be sub-menus. There are even sub-menus attached to what are primarily data entry screens. The software is written entirely in a proprietary version of BASIC that supports chaining.[3] The features are spread through more than 200 medical system specific programs. There are numerous programs that are shared by the medical and dental system. As well, many files are client specific customizations. The standard application is highly customizable. For many clients customization is handled with a large number of customization flags that are not always mutually compatible.

The software is in a relatively constant cycle of new feature development and maintenance. New feature development is often the result of requests from existing clients. When enough new features have been added to the software they are integrated into a complete system and that new system is declared to be the next version. This new version is then offered to any client for minimal cost. A client is never forced to upgrade to a new version of the software.

The integration of new features leads to problems when the new features conflict with existing customizations or pre-existing default behavior. The solution to the conflict is to create a customization flag that can be set or unset in a customization file. The code then executes one of several incompatible features by reading the customization flags and branching as appropriate. The most difficult part of upgrading a client is setting the flags in the configuration file.

This development strategy means that many different versions with different customizations are in the field and supported all at the same time. Developers and technical support staff must be especially sensitive to the version and the customizations that have been applied when helping a client, fixing a bug, or adding a feature. In this world, in this software, what appears to be a very simple problem could turn out to be a very complex interaction.

As a relatively simple example, a client had paid for a new feature which worked for a couple weeks and then mysteriously stopped working. Kelly, the client manager, was providing support and needed to solve the client's problem. She checked the documentation and found that it described two possible behaviors for this section of code. Neither behavior covered the new feature created for this client. Kelly asked the Support manager if he knew anything about the two customization flags. The Support manager told Kelly that Brad had made the change and that she should check the work order for the change. She saw that Brad was the appropriate developer and that he had described a new setting for the flag. Kelly then went to Brad's office and asked him whether the third setting described in the work order was correct. Brad examined the code and verified that for this specific client there were now three possible flag settings and the third was correct. He apologized that this was not in the authorized documentation and pointed out that this feature was not yet part of the standard version.

The problem and solution process described above is typical around MSC. They rely on artifacts and people in the environment to help them find the expertise necessary to complete their work.

EXPERTISE LOCATION

In the following sections, we distinguish two steps in finding expertise within organizations, separating finding into identification and selection phases. *Expertise identification* is the problem of knowing what information or special skills other individuals have. The above example highlighted the problem of figuring out who has what knowledge or special skills. *Expertise selection* is appropriately choosing among people with the required expertise. If there are multiple potential experts or people with the requisite expertise, it is necessary to select one (or more) to ask.

We make this distinction analytically, recognizing that it does not hold all of the time or for all participants. We wish to emphasize that expertise identification and expertise selection are iterative, interwoven behaviors at MSC and most likely in general work life. Yet, this distinction has grounding in theory, data, literature, and design criteria:

- From the perspective of s-dcog theory, solving many problems collaboratively requires first identifying the necessary resources in the environment and then obtaining them (although this may be done iteratively). Sometimes, participants will do these two activities consciously; at other times, the activities will have become routinized.

[3]In this version of BASIC, chaining is an overlay technique that allows two BASIC programs to share variables. This whole approach can be considered similar to a procedure call, but without the ability to automatically return to the prior execution context.

- Many, but not all, of our informants separated identification and selection. Many informants appeared to separate out, most of the time, how they knew people had expertise from the act of picking people they wished to approach. Informants did so spontaneously over several interview questions and in informal interviews. Even those participants who were somewhat inarticulate about identification had noticeably nuanced and articulated selection criteria.
- Paepcke [15] suggests this separation in his description and analysis of contact brokers.
- As well, a separation such as this holds promise for the design and implementation of potential systems.

The distinction between the two is valuable analytically and will prove useful in the following discussion. We cover expertise identification and expertise selection in turn.

EXPERTISE IDENTIFICATION

Identifying people who have expertise to share is a crucial first step in solving many problems at MSC. Expertise identification is a difficult problem. The nature of expertise itself, what it is and how it is used, as well as the fact that people grow and change over time, make solving this problem difficult. This is the first problem with which information seekers must deal before they can hope to get information they need.

Everyday Expertise

Many people at MSC, especially those who are senior, have difficulty articulating how they know who has knowledge about system components, diagnostic methods, business practices, and the like. For many people, "experience" is the primary guide in identifying others with specific expertise. Senior employees mention "experience" regularly in interviews and cannot articulate more detailed steps:

> You learn who's [the] most experienced in what areas. …You just know. - Sherry.

At the time, this programmer had been with the organization for more than eight years. For Sherry and others, their knowledge of what others know is part of their everyday understanding of MSC. It stems from performing work and the large gamut of social interaction, both formal and informal, that is part of that work over time.

However, junior people do not yet have this implicit understanding of who knows what. In addition, even senior people cannot track everything in even a company this small. There are, then, mechanisms that some interviewees articulated and some that we observed for identifying the right people to ask. Some of these have been observed in other studies, such as judging professional experience, organizational tenure, and geographical proximity. We describe below two mechanisms which people use at MSC, but that have not been reported in the literature. The first is how people use artifacts to gain information about who does what in the organization and the second is a specialized role that helps people find the right person.

Historical Artifacts

One mechanism for identifying potential sources is to use historical or archival data that the organization maintains. At MSC, programmers and support staff use the change history records to identify potential experts. This program change history is modified for every change made to a file in the system for as long as that file exists. Maintaining the program history is a requirement of the developer who modifies the source code, and this change history can be used to identify possible experts for technical support. Within MSC, this is sufficiently important that it has been given a name, the "line 10 rule".

> …you can simply look on the system at the program history. And we keep an on-line history of every change that's made to a program, and it shows the programmer who made that change. And so I can look at a program and I can not only see who was the last person who worked on the program was, but I can see who the last 10 people or 20 people or whatever who have worked on it. - Ian.

Developers use the change history similarly. Faced with modifying a program or file which they did not "know," developers use the change history to see who had made previous changes. The rationale is very simple. The developers use the program history to identify the person who most likely has the "freshest" memory of the code. In the following quote, line 10 contains the "mnemonics" of the programmer who last changed the code. In MSC, "mnemonics" (or "mnemonic") are constantly used as identifiers; they are roughly people's initials.

> When a programmer makes a change in a program he is supposed to add his mnemonic to the line and update the date. This is how we know who last changed the program. Whoever made the last change in the program is the default expert in that program. … It's close enough. The logic is that the person who spent time on it last has it freshest in memory and so they are the best person to ask a question. - Brad

However, the generic rule upon which support and development rely is not always correct. Making a small change can mark a developer as the expert even in the case where the small change was really insignificant:

> It's not always the right way because sometimes someone makes a one byte change on a program. Because you make that change you put your mnemonics on the program and then, say someone comes to you and, I don't know the program, I just made this one byte change. Go see this person, someone else. - Joey

There are other uses of historical data in MSC as well. In Technical Support, they create and maintain informal documents to which any support representative can contribute. Often individual contributions to these informal documents are marked by the contributors' mnemonics. This informal information is often scattered and incom-

plete, so one informant's final recourse is again to use the mnemonics as a means of guiding her identification.

The use of historical data for expertise identification may seem logical, common, even simple. But historical data presents several problems for expertise identification. For example, the use of the change history as an expertise identification technique is a very simple, effective rule of thumb that can fail. The quote by the developer above demonstrates a major problem: It is very difficult to detect the difference between small changes and larger changes with the change history. This problem can lead to falsely identifying another person as a possible expert.

Relying on the change history can result in other problems when identifying possible experts. A developer is sometimes interested in identifying other developers who have worked with a larger portion of the code. Because the change history does not effectively convey the size of a change and because the history is organized on a file by file basis, the developers must rely on some other identification technique when they are going to make a large scale change that will affect several files.

MSC staff use the line 10 rule and other sources of online information because they are the most current and accurate representations of what people are doing. They are not perfect, but they are the best available. Mechanisms such as these must be either consciously maintained or be created as an ongoing byproduct of the organization's work activity. Online information at MSC that is not constantly updated, and therefore accurate, is eventually ignored. For example, a problem solution index that was used by Technical Support fell into disuse because, in the words of one interviewee, "I don't know if someone is validating the solutions."

Similarly, in 1990 MSC experimented with the development of a formalized experts directory. The directory was generated by providing department managers with an open-ended list of topics specific to MSC technologies. A manager then filled in the mnemonics of individuals who the manager felt knew something about the topic. By most accounts this formal directory was not maintained and not frequently used. At the time of our study, the directory was seven years old, had never been updated, and was not in use.

Expertise Concierges

Organizations often have key people who have very strong very elaborated social networks. Allen's [4] discussion of the technological gatekeeper presented a highly connected organizational role which specifically served to bring new technically relevant information to potential information seekers.

The technological gatekeeper is an important organizational role that mediates many requests for information. Other researchers have found variants of the gatekeeper. Ehrlich and Cash [6] found an information mediator who facilitates access to relatively local corpus of documents and similar information. Paepcke [15] found a role similar to that of Allen's gatekeeper; Paepcke's contact broker knew people in other parts of the organization. We would expect many variants of gatekeeping, since according to s-dcog theory, this role should become specialized as it becomes adopted in specific organization's work processes.

At MSC we found another variant, critical to the information processes at MSC, the *expertise concierge*. The expertise concierge is a critical identification resource for individuals who are seeking expertise. The expertise concierge facilitates the exchange of information by referring people looking for information to those who are most likely to have that information. The expertise concierge maintains a sophisticated map of the individuals in the organization and what they know. When a person who is looking for expertise asks the concierge about people who may be able to help, the concierge shares a portion of this expertise map, identifying possible candidates. Concierges can do this because they have a high level of technical competence, allowing them to make appropriate judgements about the topics and ranges of others' expertise. Concierges both maintain their map and share their map through the social interaction of making technical referrals.

In the following quote, one expertise concierge is emphatic about his role. He considers himself an expert at knowing who knows and who does not know the answers. In the quote an explicit example is made of a particular piece of code, but the expertise that this concierge had was not just with respect to code.

> ... And what I, I am the expert in, I am the expert in knowing, I'm the one that knows who has the answers. ... Because I worked with these people from day one. And I know what they know, what they don't, what they worked on, and I can, and I see a program and I see this line of code and I look at it and I say, come on, Mary wrote this in 1992. I *know* that [emphasis in the original]. - Eyal

The quote illuminates another dimension of the expertise concierge. The type of knowledge, which these people dispense, is also a specific type of expertise. This concierge specifically considers himself an expert at this. For him, there is a type of domain expertise which people leverage to solve their problems and a type of social expertise that he dispenses to help people find those domain experts.

There were two different people at MSC who performed the role of the expertise concierge.[4] Through our observations and interviews, we came to the conclusion that the roles of expertise concierge, information mediator, and technological gatekeeper are distinct, but they share many of the same characteristics. People in these roles were tech-

[4]In fact, our use of the term is derived from one interview where the informant used the term "concierge" to describe the behavior and role of one of these expertise concierges.

nically sophisticated, had relatively long tenure with the organization, and had high-status positions in MSC.

Identification in Design

We observed varying identification mechanisms at MSC; yet, it is clear that no system can hope to implement all of them. The expertise concierge is clearly beyond current system capabilities. However, some of the identification mechanisms may form the basis for a user assistance strategy. For example, one mechanism reported here, the reliance on historical artifacts, seems promising. Historical artifacts are used in a particular style at MSC, but clearly historical artifacts vary among organizational cultures, work processes, and system goals. An assistance strategy may wish to consider the types of historical artifacts that are employed by local users as resources and then incorporate use of those within a system.

EXPERTISE SELECTION

Identifying people with expertise is not sufficient to solve one's problem or answer one's question. A person must then go to one or more people to get help. This leads to the next phase, expertise selection.

Given several different methods of identifying possible sources of expertise, the next problem is that of selecting one or more appropriate sources. It is unlikely that any single individual will have complete understanding of a very complex system. Any view of expertise that results in only one possible expert on any given topic is not very elaborate nor realistic. Information seekers are often faced with choosing from among several possible experts, each of whom possibly has the necessary expertise. Additionally, with a complex problem, seekers will likely query more than one person, collecting important insight into both the problem and the developing solution with each social interaction.

Through our interviews and observations, we were able to identify three general expertise selection mechanisms: organizational criteria, the load on the source, and performance. Informants were surprisingly detailed and nuanced about these mechanisms (especially considering their reliance on "experience" in identification). The nuance results in a long list of behaviors, each of which fit in one of the general categories.

The many variants which we found testify to the social and psychological complexity of expertise selection in practice. Selection cannot be distilled down into one or two behaviors, as will be seen in the following description. Instead, expertise selection is achieved through combinations of many, slightly different, behaviors each adding to an individual's judgement about the appropriateness of one or more expertise candidates.

The following section describes the many selection behaviors observed at MSC. While this list represents the common behaviors at MSC, we do not believe that this is an exhaustive list of all possible expertise selection behaviors.

Organizational Criteria

Organizations have norms about which they structure their members' activities. In choosing people to ask in MSC, participants kept to a standard set of rules-of-thumb. Senior personnel tended to ignore these more frequently than did junior staff, either because they had more sophisticated selection criteria or because they were "grandfathered" into an earlier and less restrictive organizational structure.

Keeping it local

The first rule-of-thumb is to keep the problem as close to the place where it originated. At MSC, as in many hierarchically-structured organizations, people prefer to stay within the organizational lines as long as possible. Developers attempt to keep within the development department, and support representatives attempt to keep within the support department.

This rule-of-thumb results in the selection of people who are most likely to understand and relate to the context of the problem. In many cases these are relative peers or immediate supervisors.

Cross department

When a problem comes to the person and he cannot solve it, the next step is to cross departmental boundaries. The following quote shows a two-step process. If the problem was primarily one concerning a support problem, this technical support person stays within Support. A peer or manager is completely acceptable. If the problem concerns the programming of the system, he goes to Development.

> … And if it's related to the project itself I go straight to the project manager [in my department] first. … [Instead] say it's a programming question that is also related to the project, but is more a programming issue, I'll ask one of the programmers that's available … - Mario

This process is echoed by the Technical Support manager:

> If you're asking me that, then depending upon what in particular, in this case we're talking about something that looks like it's program related, I would go generally to the expert in that program. - Ian

Note that the Support manager goes directly to the person in the other department whom he thinks is the expert in the specific area. During the interview Ian was asked why he did not go to the other manager. He replied that the Development manager would probably not have the immediate knowledge of the program to facilitate a solution and that using the chain of command would only slow down obtaining a solution.

The last resort

At MSC, there is a final place for all technical questions. Above we have discussed the expertise concierge. One of these is assigned the position of internal technical consultant, with the expectation that this consultant will get an answer or develop a solution if there is no available expertise:

> … Because people ask me question every time I have to have an answer. … Right. See, I have no where to go. I have no where to go. … The code, yeah, that's it, you see. … - Eyal

Eyal often ends up reading the code to find solutions to problems. The internal consultant represents the last resort for all technical questions and problems.

Load on the Source

When selecting a person our informants often considered the person's workload. This load results from both regular day-to-day work activities and the added burden of serving requests for expertise. It seems that both the immediate workload and the workload over time are important criteria in selecting a person to pursue for help.

Selection based on regular workload

Rarely is it the case that an organization allows a person to sit idle waiting to be asked questions. Organizations hope to get the most out of each employee. It is likely that when looking for information, an information seeker will find some experts busier than others. At MSC, one criteria used in selecting is the workload that the expert must handle in the performance of their day-to-day duties. A junior level developer stated his rationale for careful preparation and selection this way:

> ... The people you ask have work to do too. They aren't just sitting around waiting for you to ask them questions. - Han

Technical Support has an institutionalized mechanism to estimate one another's workload. An in-house help management application maintains a "call list." The call list tracks all recent support requests made by all clients assigned to a specific support representative. Any support representative can view any other's call list by simply entering the mnemonics of the desired representative.

The length of an individual's call list is important shared information. A junior support representative, who frequently relies on the expertise of her senior colleagues, put it this way:

> … And I also, sometimes I look at their list, I go who's busy today, who has the most calls. … - Baht

This support representative uses shared information that indicates the general workload of other support representatives to decide who will be approached first. The lower the workload the more likely she is to approach them for help.

Developers do not have a similar system to compare the day-to-day workload of the individual developers. The development group has a system that pairs individual developers with specific work orders, but this system cannot be easily used to compare the workload of different developers. Instead the amount of load on any given developer is largely gained through word-of-mouth, closed office doors, and co-workers' assessment of one another's workload.

Selection based on workload over time

Day-to-day workload is not the only consideration. People also select an expert based on the workload imposed over time. That is, a person wanting help will try to disperse the questions among the possible experts in an attempt to distribute the effort, so as not to be seen as a pest, annoyance, or worse. The same junior support representative not only considers how busy her senior colleagues are on the given day, but how many times she has approached them for help:

> …I don't want to bug them too much. … - Baht

At MSC, assigning senior personnel to train new people for approximately three months ameliorates this issue.

Performance

The way experts share their expertise influence how and why they are picked, or not picked, by people who want their expertise. Sharing expertise is a type of performance [8, 14]. For the MSC participants, different experts have different strengths and weaknesses, and people choose experts based on the nature of the problem and an expert's performance characteristics. As well, within MSC, performance is often visible, and other employees note the personal and professional characteristics that are made visible as someone communicates and interacts with those who ask for help.

Problem comprehension

Problems that require expert help are often difficult problems. The very nature of the problem may itself be difficult to describe. In the best situations, an expert will engage in a dialog with the person seeking help to get a rich, fuller understanding of the nature of the problem. This does not happen all of the time. There are times when a problem is difficult to explain to the expert because of cultural differences, language problems, or a lack of experience in a related but necessary discipline.

The following quote is from a technical support manager who had worked at MSC for many years. In the quote he is quite clear that he feels that a certain developer is "the expert in Program SUMIO" which relates to processing insurance claims. He describes his difficulty explaining to the developer the impact that a problem with Program SUMIO has on a client's business:

> Um, Kumar is a good example. Kumar, his first language is not English and he has communication problems. … I would say he's the expert in Program SUMIO, in the medical, but sometimes to sit down and explain something to him doesn't work very well, and from our perspective it's important. … And he doesn't necessarily always understand the impact to the client and to their business and running their business. And part of that is the communication skills and so forth. - Ian

To work around Kumar's difficulty in comprehending the business-related problem, he instead goes to MSC's internal consultant. The manager explains the business problem

to the consultant who then translates and explains the problem in technical details to the developer.

> And so sometimes I will go to Eyal and explain in a procedural manner or in a business manner, here's what it means to the client, here's what they are trying to accomplish and then he can translate technically to him (laughs) what he needs Kumar to go into the program and do or fix or, or whatever. - Ian

Communication and language skills are only part of the difficulty that the support manager has with this expert. It seems that Kumar also does not have an understanding of the nature of the client's business. In this case, Ian cannot use the expertise of Kumar easily.

Providing a suitable explanation

Some experts are just better at providing explanations of the problem and the solution [1]. That is, the way in which an expert provides the explanation is another performance criterion which people use when selecting an expert:

> ... There's certain people where, um, they, every time, they explain things very well to me and it just takes one shot and its explained; I understand it. But, um, you know those people sometimes are really busy and it's hard to get a hold of them. ... Those times I go to others where, um, I can go any time but sometimes … I have trouble understanding the way they explain things to me. - Baht.

Like others, she prefers to pick people who provide clear, understandable, explanations. When a preferred source is overloaded, Baht reluctantly goes to people who do not share or explain effectively.

Attitude

Problem comprehension and solution explanation are not the only performance criteria people use when selecting an expert. People who look for help often mention the attitude of the expert. In informal discussions people are clearly reluctant to go to an expert who has a "bad attitude":

> ... there are people that you know that know a lot of stuff, but they won't give you information … its just they're that type of person … - Terry

Often this bad attitude is accompanied by an example where the problem or the person seeking help from the expert was belittled, treated with contempt, or even made into a joke.

The fact that a person cannot easily understand the problem, is poor at diagnosis, cannot explain the solution, or is labeled as having a "bad attitude" does not completely rule them out as a possible source. The expertise is still desirable, and in some cases it is essential. In these cases people who need the expertise will still pursue (perhaps reluctantly) the person with the needed expertise.

Selection in Design

Like identification, it is not likely that all selection strategies can be automated. In fact some selection strategies (e.g., considering an expert's attitude) will be very difficult to automate. Strategies like observing workload or following organizational structure that guide selection and reselection look like promising candidates for possible designs, since they can be estimated computationally by obtaining work data. Even still, workload is difficult to measure directly and social norms, like closing one's door, provide powerful cues to information seekers. Selection assistance should minimally consider alternative, implicit representations of workload as well as a mechanism for the individual to mark himself as busy. Again, we note the utility of augmentative systems; these would help users identify and *narrow* (rather than completely select) potential candidates for expertise.

ESCALATION

There are times when a person will make a selection, approach an expert for help, and not get any help. The identification or selection mechanisms have failed in some way. Escalation is the way in which people repair failures in identification and selection.

A simple example of escalation occurred when Mike, a support representative, helped a client with "payment aging" (used to handle late and deferred payments). MSC's software supports two aging methods, and "nearly all" of MSC's clients use the same one. This client called Mike because payment aging had apparently ceased to work. Using diagnostics, Mike determined that the client was trying to switch from the standard to the non-standard aging method. Mike could not figure out why aging was not working, so he checked the change history. He saw that Han had recently worked on that code and went to Han for help. However, Han had made only minor changes to the code and was not sure why the client was having a problem. Han and Mike then went to ask Eyal for help, but Eyal was on vacation and the problem languished waiting for Eyal's return. When Eyal returned, Han went to him with the problem. Eyal quickly emphasized that that client with that configuration should use the standard method, closing the problem.

The above example shows two different breakdowns and a simple escalation that eventually solved the problem. In the first breakdown, Mike used the change history to identify Han as a likely candidate for the necessary expertise, but the change history was misleading in this case. This was a breakdown in identification. In the second breakdown, Han correctly identified Eyal as having the expertise to solve the problem (Eyal being a concierge). But the selection failed because Han did not know or had forgotten that Eyal was on vacation and unavailable.

As a generalization of this example, expertise identification can fail in three ways: over-identification, under-identification, and misidentification. Over-identification provides a set of candidate sources that is entirely too large. The set includes people who have the necessary expertise but it also includes a large number of people who do not. Or, it may include people who are truly experts when this

level of expertise is not required. Over-identification is likely to result in the selection of a person without the proper level of expertise.

Under-identification provides a set that is too small. With under-identification, all of the identified candidates have some expertise in the necessary topic, but the set excludes at least one person who is necessary for the problem to be solved.

When only one or two people are required, under-identification is the same as misidentification. A misidentification breakdown identifies a set of people, but none of these people has the required expertise, at a sufficient level, to solve the problem or answer the question.

Expertise identification is not the only place where a failure might occur; a breakdown may also occur in the selection phase. For example a breakdown in the load balancing strategy may result in selecting someone who is actually too busy to respond. Alternatively, a breakdown in using performance criteria may result in the selection of a candidate who just cannot understand the nature of the problem.

Escalation provides a way to either adjust the set of candidates initially identified or to reselect from among those candidates utilizing information gained in the initial attempt. (We note that just as identification and selection are occasionally only analytically separable in the initial search, this is also true in escalation.) The term escalation is used here to describe an increase in the intensity with which a possible solution is pursued. Escalation does not always mean that a problem is pushed up or down the organizational hierarchy. Within MSC, people seeking information may go to less desirable sources (e.g., to people with less expertise or to ill-maintained documents), sources with a higher psychological cost (e.g., to objectionable people), or cross departmental or even organizational boundaries.

In practice, breakdowns and their repairs are fluid. The following example shows how breakdowns in identification and selection, as well as their repairs, are interwoven and fluid. Although this was an unusual event, it highlights critical features of escalation at MSC.

The Server Crisis

MSC had been having problems with a high-speed dedicated line between their headquarters and a machine across town that was their gateway for all outside connections. Late one Friday afternoon, the substitute system administrator ("sysadmin") decided to work on the problem, the normal system administrator being on vacation. The substitute changed a few fields of the network configuration file, ran a test, and saw that the throughput between the warehouse and headquarters had not changed. Since his attempted fix had not changed the problem, he reset the configuration back to what it was when he started. Thinking that everything was working, the substitute sysadmin left for the weekend.

Monday morning a senior developer, Karl, was called at home by a client who asked why the system was down. The senior developer logged on and checked the processes on the network server. He quickly decided that this was a serious problem:

> At that point I got on my PC at home and tried to get in. I could see there was a problem. Well, I telneted and I could see, no processes, that nobody could get in. I decided that the problem must be critical so I paged Daniel [V.P. of Development]. I had Laurie call or page Andreas and Craig. - Karl

The call to Daniel is an example of tightly-tied identification and selection. Daniel is very knowledgeable, but in this case Daniel's organizational status was more important. Karl called Daniel first, not to get explicit help, but to get recognition that the problem was critical and would require additional resources.

At the same time, Karl requested assistance from Craig and Andreas. Karl requested that they both come to work early. In this step, Karl specifically marshaled expertise resources that he wanted at his disposal: Craig, an expertise concierge, was a manager, and Karl was hoping that Craig would decide to come in earlier than normal. In effect, Karl identified Craig as having important expertise if Karl and Andreas were to fail in their initial attempts.

Andreas was already at MSC when Karl arrived. Together Karl and Andreas, both programmers, tried various diagnostic routines. They checked the router link, called their dedicated line provider for a line diagnostic, and even checked the network card in the server. Lastly, they restarted the server. Everything checked out, but the server was still not accessible from the outside. Karl then turned to Craig for help.

As mentioned, the identification of Craig occurred earlier, but his selection had been deferred. Craig was not consulted immediately because Craig was in a different part of the organization. Instead, Karl selected Andreas because of Karl's desire to "keep it local." This turned out be fruitless; Karl and Andreas exhausted everything they knew to try. An escalation then occurred where Craig was explicitly used as a concierge to suggest others who could help. In the escalation, Craig suggested the substitute sysadmin and an off-site employee who was very knowledgeable about Unix. In this second escalation, the substitute sysadmin was quickly selected. This provided the solution:

> We got on a phone conference with a guy out in the Edinger building [across town]. He used the Unix setup program. I was unaware that Unix had a setup program. Among all the options, one was changed. I don't know what these options do, but I noticed that something was missing. The default gateway was missing. I don't know what it was but it's not workable this way. The guy … had a screen shot from last Friday and he confirmed that it was missing. I don't

know why he had a screen shot. But putting that [the default address] back fixed the problem. - Andreas

This example shows escalation as a standard repair technique at MSC. In simpler cases, like the first example, the escalation would likely end with an expertise concierge, Craig or Eyal. Indeed, in the end, the substitute sysadmin was finally included at Craig's recommendation. The example also shows how participants sometimes separate identification and selection of the necessary expertise: Karl reserved the use of Craig. As well, the substitute sysadmin was selected in the second escalation, after two potential candidates were identified.

Escalation in Design

The above example demonstrates that breakdowns occur naturally. Expert-locator systems are likely to have breakdowns too. As with identification and selection, escalation behaviors may be too difficult to support fully. It is not clear that the reservation of expertise, as Karl did with Craig above, can be supported through a system.

Escalations, except in the simplest cases, will have multiple iterations. Multiple iterations require mechanisms for tracking a problem and its state, both social and informational. On any iteration, an expert-locator system will need mechanisms to understand what a user has previously attempted so that suggestions can be modified to fit the severity and situation of the problem. This argues for at least two types of feedback and modification techniques; one that can handle the escalation of a single problem over a short period of time and one that can tailor itself to the preferences and needs of an individual user.

SUMMARY

Our field study of information sharing at a medium sized software company focused on the expertise that is leveraged through social interactions by the participants. We observed participants solving two general finding problems through social mechanisms: expertise identification and expertise selection. We believe that these are two crucial problems that must be solved for individuals to satisfy their need for expertise. We believe that the escalation behavior that we describe here also represents an important expertise location mechanism.

Our goal with this field study was to provide the broader research community and ourselves a rich, empirically based description of expertise sharing that can be the basis for new expertise sharing information systems. The next step in our research agenda will be to pursue the development of new systems based on these results from our field study.

Acknowledgements

This project has been funded, in part, by grants from National Science Foundation grant (IRI-9702904) and the University of California MICRO program. This project benefited from the reviewers' comments and from conversations with John King, Jonathan Grudin, Kate Ehrlich, Fernando Olivera, Stephanie Teasley, Loren Terveen, Christine Halverson, Tom Gruber, Clark Turner, Suzanne Schaefer, the ICS Graduate Student Advancement Colloquium, and many others. Our research group, Jack Muramatsu, Wayne Lutters, and Keri Carpenter, contributed to this understanding of expertise and organizations. We would also like to thank the participants at MSC for their insights and assistance.

References

1. Ackerman, M. S. Answer Garden: A Tool for Growing Organizational Memory. Ph.D. Thesis, MIT, 1993.
2. Ackerman, M. S. Augmenting the Organizational Memory: A Field Study of Answer Garden. Proceedings of CSCW '94, 1994: 243 - 252.
3. Ackerman, M. S. and D. W. McDonald. Answer Garden 2: Merging Organizational Memory with Collaborative Help. Proceedings of CSCW '96, 1996: 97-105.
4. Allen, T. J. Managing the Flow of Technology. MIT Press, Cambridge, 1977.
5. Cicourel, A. V. The Integration of Distributed Knowledge in Collaborative Medical Diagnosis. In Galegher, J., R. E. Kraut and C. Egido (ed). Intellectual Teamwork. Lawrence Erlbaum, Hillsdale, NJ, 1990.
6. Ehrlich, K. and D. Cash. Turning Information into Knowledge. Proceedings of Digital Libraries '94, 1994: 119 - 125.
7. Foner, L. N. Yenta: A Multi-Agent, Referral-Based Matchmaking System. Proceedings of Agents'97, 1997:
8. Goffman, E. The Presentation of Self in Everyday Life. Anchor-Doubleday, New York, 1961.
9. Goldberg, D., D. Nichols, B. M. Oki and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 1992, 35(12): 61 - 70.
10. Hill, W. and L. Terveen. Using Frequency-of-mention in Public Conversations for Social Filtering. Proceedings of CSCW '96, 1996: 106 - 112.
11. Hutchins, E. Cognition in the Wild. MIT Press, Cambridge, MA, 1995.
12. Kautz, H. A., B. Selman and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. Communications of the ACM, 1997, 40(3): 63 - 65.
13. Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon and J. Riedel. GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, 1997, 40(3): 77 - 87.
14. Orr, J. E. Talking About Machines: An Ethnography of a Modern Job. Cornell University Press, Ithaca, 1996.
15. Paepcke, A. Information Needs in Technical Work Settings and Their Implications for the Design of Computer Tools. Computer Supported Cooperative Work, 1996, 5: 63 - 92.
16. Resnick, L. B., J. M. Levine and S. D. Teasley (ed). Perspectives on Socially Shared Cognition. American Psychological Association, Washington DC, 1991.
17. Strauss, A. L. Qualitative Analysis for Social Scientists. Cambridge University Press, New York, 1987.