Seeing What You Are Hearing: Co-ordinating Responses to Trouble Reports in Network Troubleshooting

Steve Whittaker[1] and Brian Amento[2]
Sheffield University, UK[1], AT&T Labs-Research, USA[2]
s.whittaker@shef.ac.uk, brian@research.att.com

Abstract. Real time team co-ordination is a central problem for CSCW, but previous attempts to build novel systems to support it have not been greatly successful. One reason for this is that teams have often evolved highly effective work practices involving paper. In contrast to these prior negative findings, we present an instance of a successful digital system to support real-time co-ordination. Our system is designed to co-ordinate rapid responses to serious network failures in a telecommunications company. A critical reason for our system's success is that (in contrast to many prior studies) the primary data in our setting is *speech*. The support team must co-ordinate responses to trouble reports sent in voicemail messages. Our fieldwork suggests that because speech is ephemeral and not inherently visual, existing paper practices make it hard to extract information from those messages in order to construct shared visual representations of the major elements of the work. This in turn makes it difficult to co-ordinate work. Our novel system makes visible the content of these messages along with the actions that team members are taking to deal with them. An 8-month system field trial showed that making important aspects of individual work visible enhanced awareness and team co-ordination.

Introduction

Real-time team co-ordination is a central problem for CSCW (Hutchins, 1995, Malone and Crowston, 1992). This problem is made even more demanding when there are requirements for teams to respond to rapidly changing situations. Empirical studies of air traffic control (Hughes et al., 1992), team navigation (Hutchins, 1995) transport co-ordination (Heath and Luff, 1992), managing the space shuttle flight (Patterson et al., 1999) and neurosurgery (Nardi et al., 1993) have documented the major challenges of such real time co-ordination. The complex cognitive demands of such situations mean that work is necessarily distributed among different team members often with different areas of expertise. This in turn requires that teams co-ordinate their individual work, sharing the results of their individual activities and redistributing individual work-loads when necessary.

Two important methods for achieving real-time team co-ordination are awareness of the actions of others, and access to shared visual representations. Heath and Luff (1992) document the importance of a shared physical space in affording lightweight access to, and awareness of, the work of others. Hughes et al., (1992) describe how paper flight strips serve as a shared visual representation that promotes co-ordination between air traffic controllers. The shared visual representation makes important parts of individual work visible to others -- facilitating awareness and co-ordination. Shared visual representations support critical cognitive and social functions: they both serve as an attentional focus, as well as a record of progress through the task, allowing team members to co-ordinate their individual contributions to the collective task.

But prior CSCW systems to support real-time co-ordination that have been based on these insights have not been greatly successful. One reason for these failures is the nature of pre-existing work practices. Teams have often evolved highly effective work practices that are associated with paper. These paper-based processes allow workers to exploit shared representations, making work visible and shareable in ways that are hard to duplicate using digital media (Hughes et al., 1992, Sellen and Harper, 2002, Whittaker and Schwarz, 1999).

Figure 1 – The Network Operations Centre

In contrast to these prior negative findings, we present an instance of a successful digital system to support real-time co-ordination. Our system is designed to co-ordinate responses to serious network failures in the Network Operations Centre (NOC) of a telecommunications company, where complex problems must be diagnosed and responses co-ordinated within minutes. We will argue that a critical reason for our system's success is that (in contrast to prior studies) the primary data in our setting is *speech.* The support team must respond to trouble reports that are sent in as voicemail messages. Our fieldwork suggests that because speech is ephemeral and not inherently visual, it is hard to construct shared visual representations of the major elements of the work using paper-based processes. This in turn makes it difficult for the support team to co-ordinate work. This emphasis on co-ordinating *spoken* as opposed to textual information is consistent with the focus of other recent successful CSCW systems (Hindus et al., 1996, Patterson et al., 1999).

The structure of the paper is as follows: we first describe the setting for our study, the Network Operations Centre, and the work that gets carried out there. In particular, we describe 3 months of fieldwork documenting the co-ordination problems resulting from the use of the existing voicemail system. We next describe and motivate our system that was designed to assist with processing and co-ordinating responses to incoming network trouble reports. We report the results of an 8-month field trial of the system in operation, documenting its successes and failures. We conclude with a discussion of the theoretical and practical implications of our study.

The Network Operations Centre: Setting and Fieldwork

The Network Operations Centre

The setting for our work was the Networks Operations Centre (NOC) of a large telecommunications company (BigTel). We made multiple visits to the NOC over a three-month period, interviewing various personnel and observing NOC operations in order to understand the operation and demands of network trouble-shooting. We interviewed 15 support and technical engineers as well as their managers, and carried out 8 days of observations of NOC activities. All interviews were conducted during low tempo work periods. The fieldwork constituted about 100 hours of interviews and observations, and this provides the basis for the new system we implemented and evaluated in the NOC.

The goal of the NOC is to control the running of large numbers of national and international data networks, in particular to diagnose and repair network failures. Network failure is particularly costly to BigTel, because the company negotiates contracts with large companies that guarantee very high levels of trouble-free operation. These contracts contain penalty clauses, in which BigTel has to compensate these customers for even short periods of network malfunction. And a single malfunction may affect several large customers. This means that network failures need to be addressed within minutes of their detection.

The NOC is a complex open-plan working environment resembling a “warroom” (see Fig. 1). It contains about 40 office cubicles each containing multiple computers and telephones. At any one time, about 20 of these offices are staffed by managers, technical and customer-support engineers. As Fig. 1 shows, the room is about 15 metres high, and one wall is entirely taken up with large screens displaying various types of data that are relevant to predicting and diagnosing networking problems. Some of these screens contain detailed information to help engineers address emerging networking problems, e.g. networking node diagrams showing routing information and network load in the BigTel network. Others contain information about incoming customer reports concerning recently detected faults in the network, or roster information about which engineers are manning the current shift. Yet other screens provide information about news and weather. These are important because cataclysmic news events, such as earthquakes or terrorist strikes can have large effects on network traffic loads, vastly increasing the likelihood of failure.

There are three types of personnel in the NOC, technical engineers, support engineers and managers. The role of the technical engineers is to anticipate, diagnose and repair networking problems. The role of the support engineers is to interact with customers, both to interpret and evaluate incoming customer reports of networking problems, to determine their severity and importance. If customers report severe problems, support engineers may escalate these, so that they are dealt with immediately by a critical response team. Customer support engineers also interact with customers about the current progress of the technical engineers' repair efforts and their likely outcome. Engineers work 8-hour shifts and on any shift there are about 10 technical engineers and between 2 and 5 customer support engineers.

In what follows our main focus is on support engineers and their work processes. Over previous months they had experienced some problems in executing their work. Our remit was to identify possible reasons for their problems and to devise software that might address these.

Customer Support in the NOC

BigTel receives many reports of network failure through a web-based system, allowing customers or account representatives to enter a description of their networking problem. These failure reports are known as BMP (Business Management Process) reports. Most problems are taken care of using this system. A technical engineer will pick up a BMP report, fix the problem and then close the BMP ticket.

However a second, more important, set of failure reports are received in voicemail. These are the sole responsibility of the support engineers and are known as "trouble reports". Important customers and account representatives are given a direct support hotline that they can use to escalate problems that they feel are not being dealt with quickly enough using the online BMP system. Customers may also leave voicemail when networking failures mean that they cannot access the web to register their problem - making the phone their only possible communication mode. If the customer problem is indeed serious, or if multiple customers are reporting similar problems suggesting a large-scale network failure, the support engineer may then decide formally escalate the problem. Escalation leads to the immediate creation of a critical response team of 3-10 technical and support engineers who are standing by for such a situation. The team's goal is to diagnose and repair the escalated trouble immediately.

It should be noted at this point that escalations are unusual. Most trouble reports do not require escalation. There are highly effective automatic diagnostic processes in the network. This often means that by the time that the trouble is formally reported by the customer, the technical engineers have already fixed the problem, or are in the process of doing so. Nevertheless, the support engineer's role is important: (a) to escalate when necessary, (b) to identify those few cases that have not been reported elsewhere ensuring that appropriate remedial action is undertaken, and (c) for more usual cases to report back to the customer that their troubles are being dealt with.

The decision to escalate depends on a number of factors, including the seriousness of the problem, the number of customers reporting it, and the identity of the reporting customers. Often the decision to escalate depends on the number of incoming trouble reports. Networking failures seldom come in isolation, so that if the support engineer detects that multiple incoming trouble reports are related, this may be cause for escalation. For example, a major failure of the network can mean that multiple customers in one part of the country will suddenly all report concurrent problems. In order to detect such patterns, the customer engineers need to be aware of the details of the trouble reports that their colleagues are currently processing, along with other more general information provided by the wall screens shown in Fig. 1.

If the customer support engineer decides to escalate, this process involves setting up two different conference calls: one used by the escalation team to diagnose the technical repairs of the network and the second to interface to the customers. The technical discussions are often fraught and complex, and the aim is for the support engineer to serve as a conduit between customer and escalation team on these two separate calls, relaying relevant information between them.

The final job of the customer support engineer is accounting for their activities. At the end of their 8-hour shift, each support engineer generates an activity summary. The support engineer must state: which trouble reports they processed, when they processed them, the content of each report, and the action taken in response to the call. These activity summaries are important because they help BigTel to determine responsiveness to customer problems. Sometimes there are requirements to follow up on a previously dealt with call (*"What did we do about customer X's problem?")*, and these reports are the main source of information for such follow up activity. It is also important for the company to be able to determine the precise time a call was received, along with the exact details of the remedial action. This enables the company to determine whether legitimate problems were responded to as quickly as possible.

The critical duties of the support engineers are therefore to process incoming calls, to decide for each whether it warrants escalation, if necessary to co-ordinate that escalation with technical engineers and customers, and to summarise the results of their activities with respect to each trouble report.

Our interviews and observations revealed some important problems experienced by the support engineers. These stemmed from difficulties in: (a) processing incoming voicemail messages to determine what action is required; (b) co-ordinating with others about how to address the trouble reports; (c) generating reports of their actions to support follow up and audit activity.

The Problems of Processing and Co-ordinating Reponses to Trouble Reports

Processing Messages

One major barrier to the effective processing of incoming trouble reports is that these are *spoken.* Often voicemail messages contain quite complex information such as caller name and callback number, account names, times and BMP ticket numbers. And the technical details of the trouble itself may involve technical terms and acronyms. In addition, the trouble report may be unclear because it was generated under the duress of system failures at the customer site.

Here is an example anonymised trouble report:

> “this is [** - identifying information removed] GNOC, we have 198 T3s down at 0914 EST, not counting each PVC. Frame Relay state that they have failed cons between [**] and [**]. As of 0935 EST, 25 T3s were restarted.”[1]

The engineers were unanimous about how difficult these voicemail messages were to process. One support engineer noted how hard it was to determine what the message was about and what action needed to be taken. These problems were also exacerbated by the need to process messages rapidly:

> “we need to respond to those calls immediately because the network may be down but it can take several minutes to process one call because the information in there is so complicated”

Processing each message often involves replaying it several times with pen and paper at hand to take detailed notes of exactly what was said. Clearly, taking minutes to process a message is unacceptable, given the requirement that problems be addressed immediately.

Overall our informants were concerned about their ability to extract information from these messages, because of the general difficulty of extracting information from speech, the complexity of the messages, and the time pressure of needing to respond rapidly. These factors together meant that engineers were unconfident of the quality of the extracted information in their handwritten notes.

Co-ordinating with Others

In addition to these problems of extracting information from the voicemail trouble reports, a second set of concerns surrounded the co-ordination of work between support engineers.

---

[1] GNOC stands for global network operations centre, T3s are high bandwidth networks, EST is US eastern standard time, PVCs are permanent virtual circuits, Frame Relay is a type of network technology and cons. is an abbreviation for connections.

All incoming verbal trouble reports are delivered into a single shared voicemail box. All members of the customer support team have access to the mailbox, and they take turns to access the calls. Their workflow process is the following.

Given the need for immediate processing of voicemail trouble reports, it is imperative that one of the customer support engineers processes each trouble report the minute it arrives. All support engineers constantly monitor their voicemail. As soon as the "message waiting" light appears, anyone who is not directly engaged in processing other messages, or interacting with another customer picks up the new message. They then process the message, taking notes on paper as described above. They then delete the message. The reason for deleting the voicemail message is to prevent duplication of effort. Deletion stops a second engineer from inadvertently picking up the message and processing it.

The engineer responsible for the message then decides what action should be taken, i.e. whether the message warrants escalation, whether the fault is already being dealt with or whether to file a routine BMP ticket. Once this action has been carried out and the problem addressed, the engineer notes the results, and then proceeds to the next message.

Our observations suggested a number of co-ordination problems resulting from this process. The practice of deleting messages from the shared voicemail box and the inability to share private handwritten notes about messages means that each support engineers' work is largely inaccessible to others. In consequence engineers have little idea about what troubles their colleagues are dealing with. More specifically the following co-ordination difficulties arise:

(1) *Determining the number of outstanding trouble reports and who has responsibility for each.* The "message waiting" light in the voicemail box indicates that there is at least one message to be processed but it does not indicate the exact number if there is more than one message. This makes it hard for the team of support engineers to determine their outstanding workload, as one cannot see at a glance how many unprocessed messages have arrived. It also makes it difficult for support engineers to allocate responsibility for incoming messages. We observed shouted conversations between engineers across cubicles to determine whether a colleague intended to take a new call: *"Are you taking that?"*. And when there were multiple unprocessed messages, one engineer shouted to another: *"There's more than one in there. Can you get the other?"*. Such shouted conversations are not ideal because the NOC can be a noisy environment, especially when there is an escalation taking place.

(2) *Determining relations between reports and who has responsibility for each.* Once an engineer has processed a message it is deleted from the system, and its details recorded on paper. But other engineers in the team cannot see these individual notes, so they have no idea about the details of the messages taken by others. This is problematic because the decision to

escalate may in part be determined by the fact that there are multiple related trouble reports, indicating a more serious network failure. The fact that engineers cannot currently 'share notes' means that they do not have access to others' reports, and so cannot make these connections. Not knowing who is responsible for each report can also mean that workloads may end up being distributed unequally, with team members being unaware that other team members are highly overloaded. We occasionally observed engineers seeking each other out to determine who had taken a particular prior call and what it was about (*"Did you take a call from X about Y?"*). Again however, the working environment was not conducive to such conversations.

(3) *Duplicate reports.* A related problem (also concerning detecting relations between messages) arises when there are multiple trouble reports from the same customer. The customer may call multiple times if their problem is changing rapidly, or they feel it is not being addressed quickly enough. These repeat calls should logically be dealt with by the same engineer, but private note-taking can mean that support engineers may be unaware of previous messages from the same caller if these were dealt with by someone else and if these prior calls are not explicitly referenced in the current trouble report call.

(4) *Handovers of outstanding trouble reports across shifts.* Sometimes a trouble report is still unresolved at the end of the shift, but engineers have no way to publicly record these outstanding calls for the next shift. It is therefore common practice for the shift supervisor to talk to every engineer at the end of the shift, to find out the status of their calls. S/he then informs the next shift supervisor about these unresolved calls, so they do not "fall on the floor".

To summarise, the voicemail system and the current practice of taking private notes about each message leads to a number of co-ordination problems. One major difficulty is that the combination of voicemail and private handwritten notes makes the engineers work *invisible* to others. Individual engineers have no clear idea about how many new unprocessed calls there are. They also don't know which calls have been processed by which people, what these were about, or whether they have already been dealt with. As we have seen, engineers sometimes resort to shouted conversations across cubicles about who is doing what, but these types of conversations are the exception rather than the rule.

Accounting for Activities and Following Up

The support engineer's final task is to record their activities in summary reports. Towards the end of every shift they generate a detailed report of each trouble report they have handled including: caller name and callback number, account names, times, BMP ticket numbers. They also note down the main reported problem and the action that they took (escalate, call back, etc.). These summary reports are based on their earlier handwritten notes, and written up in a word processing program.

These summary reports are then combined by the shift supervisor into various daily and monthly reports containing information about the total number of reports processed, the time to resolve each report, along with details of prevalent problems occurring during specific periods. These are used for quality assurance but also to determine the validity of customer follow up queries (*'why did it take 25 minutes to address network problem X on Dec.15th?"*).

According to our informants, one problem with these summary reports is that they are laborious to construct, and sometimes inaccurate. Engineers have to type up their notes about trouble reports that they have often processed several hours ago, based on their minimal notes taken under time pressure, with no access to the original voicemail message. All the engineers complained about these accounting practices and requested tools to help them better record the facts about their calls. Supervisors were also interested in better tools here, because they understood that there were often inaccuracies in summary reports. Another problem with the reports is that they are largely generated by individuals working independently. Engineers individually type up their notes, without consulting together on cases that overlap. Again, discovering and documenting these overlaps may be important for quality control or diagnostic reasons.

To summarise, engineers and their supervisors have to compile summary reports that are used for accounting and analysis. Both engineers and supervisors are concerned about the accuracy of the information in these reports, given that they are reconstructed from fairly sketchy notes, along with complaining about how laborious it is to construct these. Finally report generation is not co-ordinated between engineers, because of difficulties of sharing notes and information about relationships between calls.

Summary

In conclusion, support engineers experience 3 major problems stemming from the fact that they are processing complex speech data:

1) It is difficult to extract complex information from speech under time pressure. As a result, extracted information can sometimes be inaccurate;
2) The combination of voicemail and private note-taking means that engineers are unaware of the work of others, i.e. what calls others have taken and what these are about. This makes team co-ordination extremely difficult;
3) Post-hoc accounting reports are laborious to construct, are felt to be somewhat inaccurate, and fail to capture overlaps between related calls.

System

System Design Goals

The system we designed and evaluated was intended to support three main tasks that users had stated were problematic in their existing set-up. It was intended to assist with:

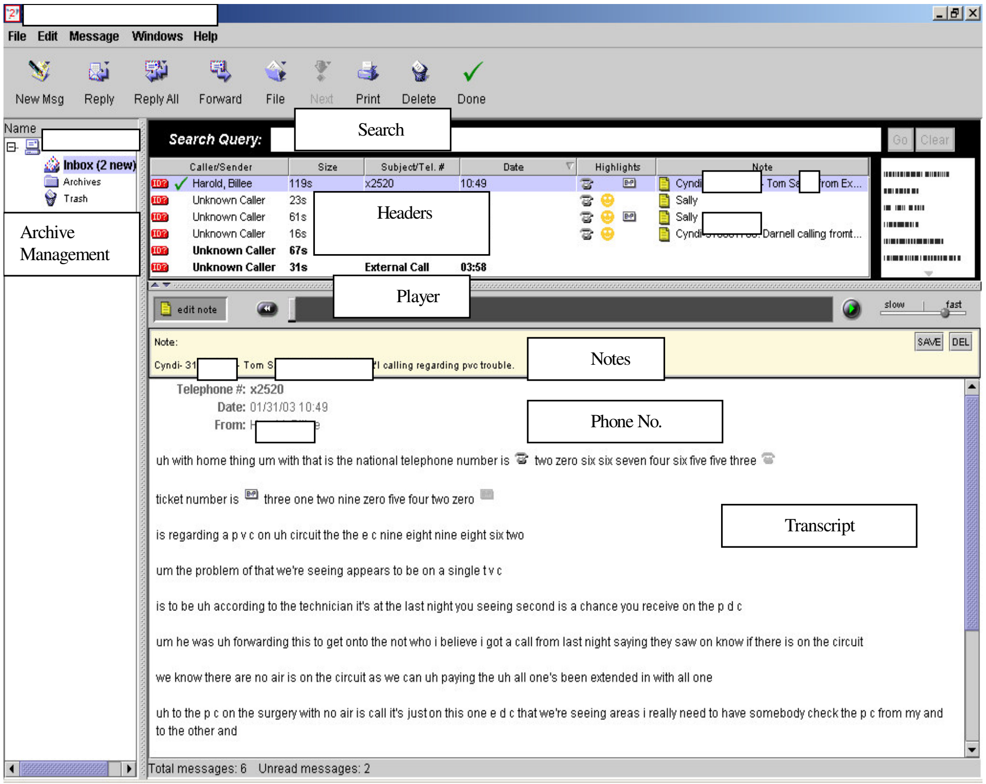
1) *Message Processing* – to improve the ease and accuracy of information extracted from speech under time pressure;
2) *Team co-ordination* – to provide a shared visual representation of the set of trouble reports to promote team co-ordination and awareness, showing:
    a. Outstanding unprocessed messages;
    b. Information about the content each message along with information about who has taken responsibility for it.
3) *Accounting and report generation* – to improve the ease with which engineers could generate detailed reports of their message processing activities, and help co-ordinate report writing.

System Design Rationale

The system is shown in Fig. 2 (with identifying information removed). Here we present the final design, but the actual system was developed over a period of several months, beginning with requirements derived from our initial fieldwork. We then iterated a series of designs with the engineers, using a variety of techniques including mockups. We worked with engineers and managers in soliciting feedback about these early designs.

The design of our system was influenced by work on both email interfaces and speech visualisation (Whittaker et al., 2002). Our fieldwork suggested that major problems with the existing voicemail system were that; (a) it made work invisible, making it hard for users to co-ordinate their activities; (b) it did not provide effective support for extracting complex information from spoken messages; (c) it offered little support for archive creation and management.

To make work visible, shareable and archivable, our system provides a visual interface to the shared voicemail box. It also allows users to take and share notes about individual messages. Our interface is similar to an email user interface in providing a header list showing important information about caller name, date, time and topic of incoming voicemail messages. More importantly it also supports access to the *content* of those voicemail messages using automatic speech transcription (ASR). Finally it provides tools for managing voicemail archives.



Caller/Sender	Size	Subject/Tel. #	Date	Highlights	Note
✓ Harold, Billee	119s	x2520	10:49	☎ ▤	Cyndi [redacted] - Tom S[redacted] rom Ex...
Unknown Caller	23s			☎ ☺	Sally
Unknown Caller	61s			☎ ☺ ▤	Sally [redacted]
Unknown Caller	16s			☎ ☺	Cyndi [redacted] Darnell calling fromt...
Unknown Caller	**67s**				
Unknown Caller	**31s**	**External Call**	**03:58**		

Note:

Cyndi- 31[redacted] Tom S[redacted] 'l calling regarding pvc trouble.

Telephone #: x2520
Date: 01/31/03 10:49
From: [redacted]

uh with home thing um with that is the national telephone number is ☎ two zero six six seven four six five five three ☎

ticket number is ▤ three one two nine zero five four two zero ▤

is regarding a p v c on uh circuit the the e c nine eight nine eight six two

um the problem of that we're seeing appears to be on a single t v c

is to be uh according to the technician it's at the last night you seeing second is a chance you receive on the p d c

um he was uh forwarding this to get onto the not who i believe i got a call from last night saying they saw on know if there is on the circuit

we know there are no air is on the circuit as we can uh paying the uh all one's been extended in with all one

uh to the p c on the surgery with no air is call it's just on this one e d c that we're seeing areas i really need to have somebody check the p c from my and to the other and

Total messages: 6 Unread messages: 2

Figure 2 – User Interface

Before deciding on this design, we explicitly considered other types of system. It is possible that more traditional CSCW systems such as shared workspaces, workflow systems or shared databases such as Lotus Notes might potentially address some aspects of the support engineers' work in allowing them for example to share their notes on-line. However these technologies have the limitation that they are text-based. As a result, they do not provide direct access to the support engineer's primary work data, namely the speech messages, nor do they offer support for extracting information from these messages. They also do not address the problem that engineers' notes can be inaccurate. Access to voicemail message content seemed to be critical for team co-ordination, along with tools for managing speech archives.

The need for access to message content and archiving argues against two other potential designs. Voiceloops (Patterson et al., 1999) supports co-ordination for the space shuttle mission by broadcasting spoken communications on multiple audio channels, allowing distributed teams to communicate and listen in on relevant conversations. However Voiceloops does not provide a *record* of conversations, so it does not support extraction of complex information from speech (which may require multiple listenings). The absence of a conversational record also means Voiceloops also does not straightforwardly support archive creation and management. For related reasons, we rejected a different design where the system simply records message ID, caller phone number, time and ID of the person responsible for the call. Again such a system fails to allow users access to message content, i.e. to provide tools for extracting information from messages, or for creating effective archives. So, while both these potential designs offer some support for team co-ordination, neither helps engineers with the analysis and management of their primary work data -- which is *speech.*

System Features

The three main goals of the interface (shown in Fig. 2 with information anonymised) are therefore to provide the team with tools for extracting information from messages, for sharing the notes that engineers took about each message they had processed, as well as creating and managing archives. The interface to the system has five major elements: header information, message body (transcript), speech player controls, note-taking, and archival features (folders).

(1) *Header information* includes caller name (extracted where possible from telephone caller ID information using reverse lookup), message length, date/time, along with message highlights (explained below) and notes that have been taken about the message (again explained below).

(2) The *message body* (shown in the bottom third of the interface) is generated by applying ASR to the message. ASR is an imperfect technology and currently ASR generated transcripts contain about 20% errors. Despite these errors, transcripts can still expedite processing of voicemail messages in the following way: users read the transcript to determine the gist of the message, but in places where the transcript is unclear due to ASR errors, they access the original speech. They access the underlying speech by highlighting the unclear region of the transcript with their mouse and then hitting the play control in the player shown in the centre of the interface. For example, the initial part of the transcript *"uh with the home thing um"* is hard to understand, so the user might highlight this and play the underlying speech. In contrast, what follows is much clearer, where the caller provides a telephone number, ticket (BMP) number, and nature of the problem (a PVC with circuit number 989862).

It might appear to the sceptical reader that the ASR transcripts are almost unintelligible. However, other research has shown that this combination of reading transcripts and playing underlying speech is an effective way to process complex speech data (Whittaker et al., 2002). Another point to bear in mind is that the transcripts are being processed by domain experts, who are familiar with both the terminology and the class of reported problems.

The transcript also contains "highlights" to help with message processing. Our interviews and analysis of the engineers' notes showed that there were certain critical pieces of information that engineers needed to extract from every message such as caller name, caller telephone number, and BMP number. We therefore used methods from AI to identify regions in the transcript where such information occurred. This is made possible because the target information has a predictable structure, e.g. a BMP number is composed of exactly nine digits. In the transcript in Fig. 2, the phone number is delineated in the transcript by the two small phone icons, and the BMP ticket number by two small 'BMP" logos. These icons allow the user to quickly scan the transcript to focus on this information, if, for example, they want to rapidly determine which problem is being reported in order to call back immediately. As with ASR, these AI techniques are not 100% accurate, but the philosophy is the same as with the overall transcripts. Users can quickly identify where in the message, the phone number, caller name and BMP ticket numbers occur. Then can then click on the relevant part of the message to play this information.

As Fig. 2 shows, we also present highlight information about caller name, caller phone number and BMP ticket number in the message header. By mousing over the relevant icon in the header, the user can quickly access this information without having to click on the message itself. We hoped that such header information would also help provide different members of the support team with important high-level awareness information about messages being processed by others.

(3) The system also supports *note-taking*. Our fieldwork had suggested that note-taking was essential to the processing of messages, and also potentially to team co-ordination. The interface allows users to type a short summary of the message in the notes field (shown in the centre of Fig 2). Notes usually include information about: the person taking the call, the caller name, phone number, BMP ticket number, nature of the problem and action taken. In Fig. 2 the notes indicate who has taken the call ('Cyndi'), the phone num ber, caller name ('Tom S.'), company and trouble description ('pvc trouble'). As with message highlights, these notes can be viewed in the message header pane, without needing to access the body of the message. One way of viewing these notes is that they provide a high level summary of message details. Our intention was that being able to see these header notes would promote awareness between engineers about who had taken which call, along with a brief summary of what the call was about.

(4) The *player controls* are standard, allowing users to select a region of the message and then to start, stop or pause playing. They also allow users to speed up playback.

(5) *Archives*. Finally, as with email, messages can be filed in the archives which operate like a standard email folder structure. And, of course the fact that we have a textual transcript for each message means that these archives (and the inbox) are searchable, allowing users to easily access specific information from the archive.

In conclusion, the system was intended to support the engineers' work processes in the following way:

1) Both the transcript and the highlight information we provide in the message headers should *expedite message processing*. While neither provides users with perfect information about the original message, they should nevertheless allow users to home in on the relevant parts of the message, playing the underlying speech where necessary.
2) The visual representation of the message inbox should make team work more visible promoting co-ordination and collaboration:
    a. The list of messages in the inbox should allow all team members to see the current workload of unprocessed messages
    b. Both the transcript, user notes and highlights should provide support engineers with access to, and awareness of, others' work
3) Transcripts, notes and the archive should assist users in generating reports, and in collaborating in the production of those reports.

Observations of the System in Use

After extensive design iterations involving the users, we installed the system and observed it being used over an 8-month period. Given the critical nature of the application, we designed our system so that it operated in parallel with the original voicemail system. This ensured that even if our system malfunctioned, the users would still be able to access trouble reports using their original system. We collected various types of information in order to evaluate the system:

1) Observations of 11 support engineers and managers using the system.
2) Interviews and questionnaires, conducted before and after the system was installed. These addressed: how easy it was to extract information from incoming voicemail messages, whether the system was effective in supporting work, co-ordination with others, whether the system had changed the way that they worked, the main advantages and disadvantages of the system compared with the prior voicemail system, changes they wanted to make to the system.

Findings

Message Processing

It was clear from engineers' comments and questionnaire responses that once the transcript was available, the combination of reading the transcript and playing relevant regions helped them to process messages relatively effectively. Once we had ironed out initial problems, the engineers usually relied on our system for accessing messages. This happened despite the fact that their original voicemail system was still available, and they were familiar with using it for carrying out their work. Users quickly adjusted to a method of reading the transcript to get the gist of message and then playing important or unintelligible parts of the transcript. They were also very positive about the extracted phone numbers and BMP tickets. Despite this improved ability to process messages, the engineers were nevertheless dissatisfied with the quality of the ASR transcripts we generated. They pointed out various acronyms and technical vocabulary that were incorrectly transcribed. Although various system iterations improved this technical vocabulary (reducing overall recognition word error rate from 30% to 20%), the engineers still maintained that they wanted absolutely correct recognition. One issue we were careful to explore was whether inaccurate transcripts ever misled engineers about important aspects of a call. Our interviews and questionnaires indicated this was not the case: engineers did not rely exclusively on the ASR transcript for message processing. Instead, as we had intended, they used the transcript to direct their attention to critical parts of the call, which they made sure they listened to. They then recorded these important aspects of the call in the notes field. Another initial problem with the system was caused by ASR processing taking several minutes to complete, which forced the users to fall back to the original voicemail system because of the requirement for immediate responding. However once we modified the system to conduct multiple ASR passes (Whittaker et al., 2002) we were able to generate ASR transcriptions almost real-time, and this promoted greater usage.

Co-ordination and Collaboration

The system was highly successful in promoting team co-ordination. The engineers quickly abandoned taking handwritten notes on paper. Instead they used the system notes facility to enter information about: (a) who was responsible for the call; (b) caller name; (c) caller number; (d) customer name; (e) problem; (f) action taken. This notes information can be seen in Fig. 2.

The engineers were enthusiastic about their new ability to see the work of others. As one engineer commented:

> ‘Now we can see who’s dealing with what call and what they decided to do about it. That helps me with handling my calls”

This comment also reveals the importance of having access to the *content* of the calls that others are processing. Engineers confirmed that both others’ notes and transcripts could be critical resources when attempting to co-ordinate. This validates our original design intuition that engineers needed to have visible access to message *content* rather than more abstract properties of the message.

The interface also allowed them to see at a glance whether there were new unprocessed messages, along with each team member’s respective workload. For example, Fig. 2 shows two new unprocessed messages and four processed messages (two of the messages are being handled by Cyndi and two by Sally). The system also promoted lightweight awareness of the actions of others. In the process of accessing a new message from the mailbox, users were able to see at a glance from header data, what their colleagues were currently working on.

The engineers were less positive about their ability to track precise message status. They pointed out that our initial system did not distinguish between messages that were currently being processed and those that were completely dealt with. At their request, we introduced message flags that could be used to signal status (e.g. the first message in Fig. 2 is ‘done’ as indicated by the tick mark (visible on the left hand side of the headers), the next three have been accessed but not yet discharged, as indicated by the fact that they are unbolded, and the final two are completely new unaccessed messages, as indicated by the fact that they are bold). This status flag was also useful for handovers between shifts. Supervisors no longer had to talk to all team members to compile a list of outstanding messages, in order to hand this over to the next shift supervisor. Instead this information was visible at a glance.

Although engineers reported benefits for the new system for co-ordinating with others, we also explored potential privacy concerns. Making people’s work more visible helps promote co-ordination and awareness, but it also makes it more straightforward to monitor various individual’s work. This seemed to be less of a problem than we had anticipated, for two reasons. First, users felt that the perceived advantages of greater visibility for co-ordination outweighed possible privacy concerns. In addition, several engineers pointed out that they already had to produce careful accounts of their activities at the end of the shift, in their summary reports, so greater accountability during shifts made little difference to them.

Archiving, Accounting and Follow-up

Finally, the engineers were extremely positive about the system’s ability to support the generation of summary reports. Instead of relying exclusively on minimal handwritten notes to write up these reports, they could refer to the transcript or replay parts of the original message. In cases where two engineers had worked on different aspects of a problem, they could co-ordinate their write-up of the trouble. And in addition, for supervisors the system made it straightforward to carry out various audit tasks such as counting the number and times of various calls. Furthermore, if there were specific follow-up questions about how a particular call or set of calls had been dealt with, then it was extremely easy to use system search to access quite fine-grained details of these messages. For example archival system search could be used to answer the follow-up question: *‘what action did we take about the persistent ** failure on Nov. 4th?”* The system allowed people to search for and access details about all prior messages, along with the engineer’s notes associated with the message. Previously they had to access large numbers of private notes and written reports to find this type of information.

Other NOC workers' reactions to the system

There were also strikingly positive reactions to the system from other people in the NOC who had observed it being used by the support engineers and their management. Overall we received extremely positive feedback about the system from both technical and managerial staff who were not part of the support team. Recall that the customer engineers work side-by-side with technical engineers, whose job is to anticipate and fix network problems. Four months after we had installed the system, the manager of the technical engineers contacted us, saying that his team had observed the support engineers using the system, and felt that it would be invaluable for their work too. He wanted us to install the system for the technical engineers. Furthermore, the NOC general manager also independently contacted us about displaying the system on one of the NOC walls. This he argued would increase the technical engineers' awareness of the workload and specific problems that the support engineers were currently dealing with. It could also inform them about potential problems that might end up being escalated to them. Together these observations provide additional evidence for the utility of our system.

Summary

Overall our system was successful in supporting both team co-ordination and archiving. The visible list of messages, along with notes and accessible transcripts made it possible for the support team to remain aware of each other's work in a way which was not possible with the voicemail system. In addition, the notes combined with transcripts and search also facilitated the process of report generation and interrogation of summary reports. Finally, general reactions to the system from other NOC workers were positive - with requests to extend the system to technical engineers and to the wall of the NOC. The system was less successful however, in supporting message processing, where users complained about inaccuracies in the automatically generated transcripts.

Conclusions

We have presented fieldwork characterising a complex real-time team co-ordination problem, i.e. dealing with network trouble reports. That fieldwork suggested that current work practices and technology made individual work *invisible* to other team members. The nature of the voicemail messages, private handwritten notes and the practice of deleting voicemail messages meant that support engineers' activities were inaccessible to each other. On the basis of these observations, we designed and evaluated a system for addressing these co-ordination problems. The system visually represented the status of incoming messages, their details, and the notes taken by the person processing the message.

Our system capitalised on prior work note-taking practices, but allowed such notes to be shared. This helped externalise individual work, making actions and decisions more visible. This in turn promoted greater collective awareness allowing engineers to co-ordinate and re-allocate work, and to take collective action where necessary. Our data also indicate that simpler systems that do not provide direct access to spoken data would have been less successful for this application.

While confirming prior observations about the utility of shared visual representations for team co-ordination, our findings contrast with other attempts to develop systems to support real-time co-ordination (Hughes et al., 1992, Sellen and Harper, 2002, Whittaker and Schwarz, 1999). In those studies users preferred existing paper-based practices to novel digital CSCW systems. How then can we explain the success of our system? One crucial feature of our system is that it requires access and co-ordination about *speech data.* In the NOC, existing work practices relied on taking (often poor quality) private notes and using the voicemail box to allocate work. These practices meant that neither the content nor the responsibility for different aspects of the work was visible. In contrast, our system capitalised on existing work practices in supporting note-taking, but critically provided lightweight access to message content and responsibility. This in turn allowed engineers to remain aware of others' work and to co -ordinate with them when necessary. These findings confirm and extend other research on audio groupware (Patterson et al., 1999) showing that novel CSCW systems can successfully support co-ordination when the primary work data is speech.

Furthermore, our approach may apply to a number of problems outside the NOC. A large number of companies have major operations to provide customer support. They often need to process customer reports and complaints that are delivered in voicemail. Many of the problems identified here seem to occur in those other support applications. For example, support teams need to co-ordinate their activity in dealing with sets of calls. The ability to make calls visible, along with supporting notes and reports would seem to be critical for this general type of task, suggesting our approach may be reasonably general.

Having said this, there are obvious places where our system can be improved. One problem is with the accuracy of the ASR transcripts. We have improved accuracy by 10% since our first installation, but clearly more work is needed to help with the difficult process of understanding and registering incoming calls. Another feature our users requested was the ability to edit transcripts directly, and we are working to provide this. We are also developing on a new forms-based user interface that will make it more straightforward for engineers to enter and view critical elements of the incoming message, such as caller name, number, problem and action taken. And in response to user requests we are also working on a large-scale display version of the interface, making it possible for everyone in the NOC to view the status of incoming trouble reports.

Finally our work underscores other research showing the importance of visual representations and awareness for co-ordination (Hutchins, 1995 Nardi et al., 1993, Hughes et al., 1992, Sellen and Harper, 2002, Whittaker and Schwarz, 1999). In contrast to voicemail, our new system made important aspects of work visible. Engineers could see at a glance who had processed which message, what that message was about and what action (if any) had been taken about it. That allowed them to co-ordinate their work with others. What was striking was how a relatively simple system was able to support these important co-ordination and reporting processes.

References

Heath, C., and Luff, P. (1992). Collaboration and control, *Computer Supported Cooperative Work*, 1, 65-80.

Hindus, D., Ackerman, M., Mainwaring, S., and Starr, B. (1996). Thunderwire: A Field Study of an Audio-Only Media Space. In *Proceedings of Conference on Computer Supported Cooperative Work*, NY: ACM Press.

Hughes, J., Randall, D. and Shapiro, D. (1992). Faltering from ethnography to design In *Proceedings of Conference on Computer Supported Cooperative Work*. NY: ACM Press, 115-122.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA: MIT Press.

Malone, T., and Crowston, K. (1992). What is co-ordination theory and how can it help design cooperative work systems? In *Groupware and Computer Supported Cooperative Work*, ed., R. Baecker, CA: Morgan Kaufman, 375-388.

Nardi, B, Schwarz, H, Kuchinsky, A, Leichner, R, Whittaker, S., and Sclabassi, R. (1993). Turning away from talking heads: an analysis of "video-as-data". In *Proceedings of CHI' 93 Human Factorsri Computing Systems*, NY: ACM Press, 327-334.

Patterson, E., Watts-Perotti, J., and Woods. D. (1999) "Voice loops as coordination aids in space shuttle mission control." Computer Supported Cooperative Work, 8(4), 353-371.

Sellen A., and Harper, R. (2002). *The myth of the paperless office*. Cambridge, MA.: MIT Press.

Whittaker, S., Hirschberg, J., Amento, B., Stark, L., Bacchiani, M., Isenhour, P., Stead, L., Zamchick G., & Rosenberg, A. (2002) SCANMail: a voicemail interface that makes speech browsable, readable and searchable. In *Proceedings of CHI2002 Conference on Human Computer Interaction*, NY: ACM Press, 275-282.

Whittaker, S., and Schwarz, H. (1999) Meetings of the board: the impact of scheduling medium on long term coordination in software development. In *Computer Supported Cooperative Work*, 8,175-205.