

From Ethnographic Study to Mixed Reality: A Remote Collaborative Troubleshooting System

Jacki O'Neill, Stefania Castellani, Frederic Roulland and Nicolas Hairon
Xerox Research Centre Europe
Meylan, 38420, France
Firstname.Lastname@xrce.xerox.com

Cornell Juliano and Liwei Dai
IDHI, Xerox
800 Phillips Road, Webster, USA
Firstname.Lastname@xerox.com

ABSTRACT

In this paper we describe how we moved from ethnographic study to design and testing of a Mixed Reality (MR) system, supporting collaborative troubleshooting of office copiers and printers. A key CSCW topic is how remotely situated people can collaborate around physical objects which are not mutually shared, without introducing new interactional problems. Our approach, grounded in an ethnographic study of a troubleshooting call centre, was to create a MR system centred on a shared 3D problem representation, rather than to use video or Augmented Reality (AR)-based systems. The key drivers for this choice were that given the devices are sensor equipped and networked, such a representation can create reciprocal viewpoints onto the current state of this particular machine without requiring additional hardware. Testing showed that troubleshooters and customers could mutually orient around the problem representation and found it a useful troubleshooting resource.

Author Keywords

Ethnography, Collaborative device troubleshooting, Mixed Reality.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

In this paper we revisit some earlier work, which proposed that a shared representation could avoid some of the problems of video – in particular the tendency to produce fractured ecologies – when used to support remotely located people interacting around physical objects. In an earlier paper [21] we presented an argument grounded in fieldwork

for why we believed that a shared Mixed Reality (MR) representation of a problem space would provide *good enough* support to overcome many of the barriers to remote participants working with a physical object, in this case troubleshooting office printers and copiers. In this paper we describe the design, implementation and testing of that system (Lighthouse).

How remotely situated people can work with physical objects is an area of interest to CSCW [8,9,14,17]. When remote interactions take place around such objects obvious problems arise from the fact that the object is not mutually shared. What are trivial matters of reference in face-to-face situations - establishing mutual orientation, understanding of referents, pointing, gesturing, knowing what people are doing or have done - become problematic when participants are remote. Various systems have been designed which make use of either AR or video in an attempt to make the properties of physical object(s) available for remote collaboration. AR systems tend to require expensive specialized equipment and although it is an area that shows promise, little in the way of working systems have yet been produced for collaborative environments such as this. On the other hand a common problem with video is that not only does it fail to recreate the richness of face-to-face interaction it also introduces new interactional problems for its users, something made evident in the early work on media spaces [10,11].

Rather than starting from the baseline of face-to-face interaction, we studied a situation where remote collaborators already work with physical objects, with just the telephone to share information and coordinate action. This enabled us to identify both what worked well and the problems of audio only communication for collaboration. As a result, the design of Lighthouse aims to *augment* the telephone rather than attempting to recreate the richness of face-to-face interaction. Given the findings of our study and the known disadvantages of video, we believed that in our situation a shared representation of the problem space could provide adequate support for troubleshooting without introducing the interactional problems of video. We therefore built a prototype MR troubleshooting system including a shared problem representation. The problem representation includes a 3D model of the machine, linked

to the machine itself – via the machines sensors – which is shared by the troubleshooter and customer. Customers and troubleshooters can talk to one another using Voice Over IP (VOIP) and interact through the problem representation. In this paper we first recall the key fieldwork findings which led to this system, then report on the system design and some user tests which provide a first validation of this system. As well as contributing to the understanding of remote collaboration, we believe this paper provides an additional contribution to the field – being an exemplar of how ethnographic field work can lead to innovative design in a real problem space, where many design constraints are practical and cost constrained.

RELATED WORK

There are two main bodies of work around helping remotely situated people to work together around physical objects – video-based systems and AR systems. Video-based systems tend to attempt to recreate the salient features of face-to-face interaction [8,9,14,17]. Studies of some of the earlier systems have shown that such systems create new environments for interaction, that is, new ecologies, since users are inevitably immersed in two environments – their local environment and the remote shared environment. Luff [19] demonstrated how conduct and ecology are reflexively related and by creating new environments with technology the relation between action and the relevant ecology may be fractured, causing interactional problems, which can make even seemingly simple activities problematic. For example, users often lack reciprocal views, making acting on objects in the local and remote environment difficult because they cannot easily design their conduct to be sensible and recognisable to the other. Various ingenious solutions have been designed to avoid these problems, for example by overlaying the helpers actions into the workers environment, e.g. through gesture [14, 15] or through drawing [23]. While these systems have had some success, they have been designed for small scale desk based tasks in static workplaces which can be easily projected to the helper and in which the helpers actions can easily be projected onto the task space. It is not clear how such systems would translate to our situation, where the workspace is large and requires the workers movement around it, or even whether there would be any value over our more minimalist system of doing so.

In the troubleshooting domain, AR systems have been created for situations closer to ours. For example, Friedrich [7] describes an AR system allowing a mobile on-site user to be instructed or to access documentation via an AR headset, in order to carry out device maintenance in large industrial plants. Bauer et al. [3] describe a reality augmented telepointer for supporting mobile, non-expert fieldworkers undertaking technical activities with the aid of remote experts. By overlaying virtual information on the real world these systems might overcome some of the problems of fractured ecologies, at least for the local party. Unfortunately they have not had the same quality of

evaluation in use as for example [19] so it is hard to know whether they solve the problems for both local and remote participants or whether they introduce new interactional problems. Certainly research has shown that head-mounted cameras can be difficult to use, particularly on the side of the helper, due to unstable and shaky views, with changing of focus whenever the worker moves his head, even to glance at the clock [8]. Indeed in the video-based systems research, greater success has been had with arm-controlled camera views, but only in table top laboratory situations [24]. Returning to the AR domain, these approaches require an Head Mounted Display (HMD), which might be envisioned to support the work of professionally-trained operators, like service engineers or mechanics [12] in high-end environments. However, in our domain of troubleshooting office devices both the cost and the required learning cannot be justified.

As an alternative to AR systems relying on video and HMD, our system takes an approach based on a 3D virtual representation linked to the device sensors. It proposes a different type of MR interaction within the continuum defined by Milgram [20]. 3D representations of devices, individuals and environments are used in various enterprise applications for which the term “Serious Games” [1] is used. Serious Games focus on illustration or simulation in several domains such as military operations [27,28], economic and business training [29], and language learning [26]. Educational and off-line simulations have been the main focus of these applications. Alternatively robotics and 3D representations have been applied to high-end environments such as nuclear plants, surgery and space operations [13,18]. The complexity of the tasks, safety-critical requirements and magnitude of the economic investment in the equipment for which the mentioned systems are designed imply the use of high-performance proprietary sensors, communication protocols and 3D rendering engines, which goes beyond our domain requirements.

METHOD

Our design method consists of ethnographic studies, from which we conceptualise innovative design solutions. We then engage in an iterative design process consisting of cycles of design and naturalistic testing. We work as a multi-disciplinary team with the ethnographers involved in the design sessions throughout the process and in which the computer scientists become immersed in the ethnographic findings. The use of ethnomethodological ethnographies in design has been commonplace in CSCW for a number of years [see for e.g. 4, 6]. Frequently fieldwork is presented with “implications for design” but rarely do we see the results of the design that is inspired by these implications – largely because such a process takes time (to illustrate our field studies were conducted in 2004). In both academia and industry there is rarely the luxury to follow a project through from studies to design to testing. On this project we

have been lucky enough to be able to achieve this and are hoping finally for product integration and customer usage.

ETHNOGRAPHIC STUDY

The field work consisted of a three week ethnographic study of a European Call Centre for a large copier and office device company. The study involved observing the troubleshooters while they worked. Field data was collected through field notes, video and audio recordings. The call centre in question provides telephone support to locations across Europe for customers with problems with their office devices (copiers, printers, multi-function devices (MFDs), etc.). Troubleshooters basic setup consisted of a PC, equipped with a call management system, a phone and wireless headset and various hard and soft-copy materials to support their work. In addition, models of all the photocopiers they supported were located around the office. In this paper we summarise the key points, more details can be found in [21,5, 22].

Troubleshooters work with the customer to collaboratively establish the nature of the problem. Although the troubleshooter has the expertise to troubleshoot the device they do not have direct access to the device or to the customer's actions. Furthermore often the customers' phone was not located near the ailing machine, causing refusal to troubleshoot, to-ing and fro-ing or involving a third party. Through talk, troubleshooters and customers work to create and maintain a mutual orientation to the device. It is this shared orientation that enables the remote troubleshooting to take place. However, this mutual orientation can break down because of: 1) the inadequate fidelity of operators' support resources, 2) the lack of mutual access to indicative resources, and 3) troubleshooters lack of direct access to customers' actions and orientation.

1) The inadequate fidelity of operators' support resources

Troubleshooters only access to the machine is through the customer. At the start of the interaction they work to establish the status of the machine and the nature of the problem, but customers are rarely experts and troubleshooters often have to translate technical terminology and reformulate problem descriptions and instructions to create a shared understanding. Customers report back on actions they have performed and the resultant machine status. In addition, when giving instructions the troubleshooters are describing sequential, physical actions to be undertaken on a real device in the absence of that device itself. They therefore have various methods for embodying the solution – such as miming the actions, going to the models of the machine on the floor, using menu maps and images. These resources help the troubleshooter visualise the sequence of actions to be performed on the device. Problems can arise because these are *generic* resources *representing* the problem device, not the problem device *itself* and thus their fidelity is not always adequate for troubleshooting. Secondly, the indicative information involved is not available to the

customer, making it a lost resource and requiring the operator to translate it into verbal instructions.

2) The lack of mutual access to indicative resources

This work of translating visual and mechanical instructions into words, and on the customers side describing what they have done and the results of it, is a form of articulation work [25] – it is largely extra work which needs to be engaged in to make the troubleshooting work in this remote setting. It is not that talk would be replaced in a completely local setting, but rather that direction and response can be an integrated mixture of the visual and verbal. Where the customer is able to locate parts easily and follow the operator's instructions, it is not necessary for the operator to be able to see what the customer is doing or where the customer is looking. However, all sorts of mix-ups can and do occur around which part each person is referring to, compounded by customers frequent lack of familiarity with technical terminology.

3) Troubleshooters lack of direct access to customers' actions and orientations

Troubleshooters need to situate their instructions in the ongoing interaction between themselves, the customer, and the device. This is a matter of parceling up the instructions to be carried out by the customers, that is, giving them in a timely manner and in appropriately sized chunks according to the customers' expertise (see also [2]). However, their resources for understanding the customers' interactions with the device are limited to what they can hear and what the customer tells them.

Troubleshooters are skilled in their work and many of the sessions pass without apparent incident, that is, where the extra verbal work required to carry out troubleshooting over the phone is adequate to resolve the problem. However, breakdowns in understanding are not uncommon and the company was keen to improve the sessions. With the key technical enablers of the machines being sensor rich and having a new user interface which could access the web, the study findings led us to believe that extra support could be provided by creating a representation of the problem space, around which the troubleshooter and customer could interact. This design is described in the following section.

LIGHTHOUSE

To address the problematics outlined above we examined ways in which the features of the actual troubled device itself might be made available to both parties. Primary here is finding ways to enable them to mutually orient to it, share indicative information, such as gesture, and enable customer actions to become available to the operator. One such way is to provide the interacting parties with a *representation of the troubleshooting problem itself*. Such a representation would provide a resource for both coming to an understanding of the problem and mutual orientation and interaction. We use a virtual representation of the ailing device synchronized with its actual status as the centre point of the representation of the problem space.

This choice was motivated by two considerations. The first motivation is to have minimal technical requirements: such representation requires only a small amount of data exchange over network and does not require any additional capture device like for video. The second motivation is to create reciprocal views which, like the telephone, give clear understanding of what does and does not fall within the shared space.

Lighthouse as shown in Figure 1, is composed of two client applications that render and control interactions with the shared representation of the problem in the device screen for the customer and on the desktop screen of the remote troubleshooter. A session management server resides between the two sites and manages the synchronization between the two clients.

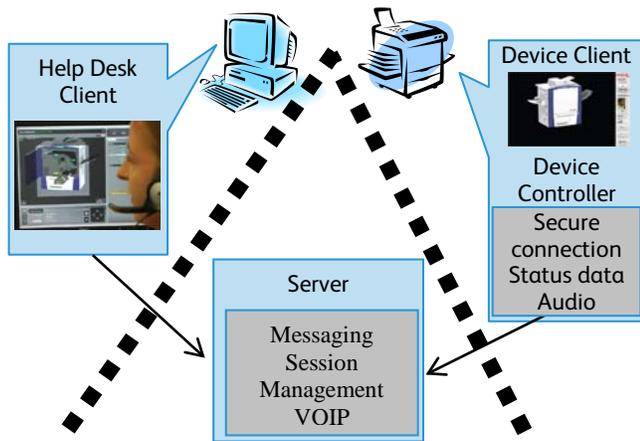


Figure 1. Lighthouse architecture.

Lighthouse has a number of features:

1. A call support button on the customers user interface to start the troubleshooting session.
2. A secure data and audio end-to-end connection to the call centre and transfer of data about the device (serial number, sensor information, etc.) to the troubleshooter.
3. A virtual model of the ailing device composed of a 3D representation of the device parts that will be visible and operable by a customer together with a semantic and kinematic model of the device that describes the various parts of the device and the way they can be operated by an end-user i.e. the various operations, states and constraints on each part.
4. The virtual model is linked to the machine sensors, so that it can reflect the status of the machine – e.g. when a door is open it will be shown as open on the model. In addition any other sensor information from the device can be communicated to the troubleshooter and displayed on their interface.
5. The virtual model is displayed synchronously for the customer on the device interface and for the troubleshooter on their terminal interface.

6. A number of means of interacting with it adapted to the user' and troubleshooter' roles in the troubleshooting task, e.g. rotating, pointing, etc.
7. A view of and access to the user's local user interface (LUI). Since this is already a virtual object it does not require modelling.
8. A VOIP connection between the customer and the troubleshooter to enable them to talk to one another.

Viewing and interacting within Lighthouse

Figure 2 and 3 show the customer and troubleshooter interfaces, respectively.



Figure 2. The customer interface.

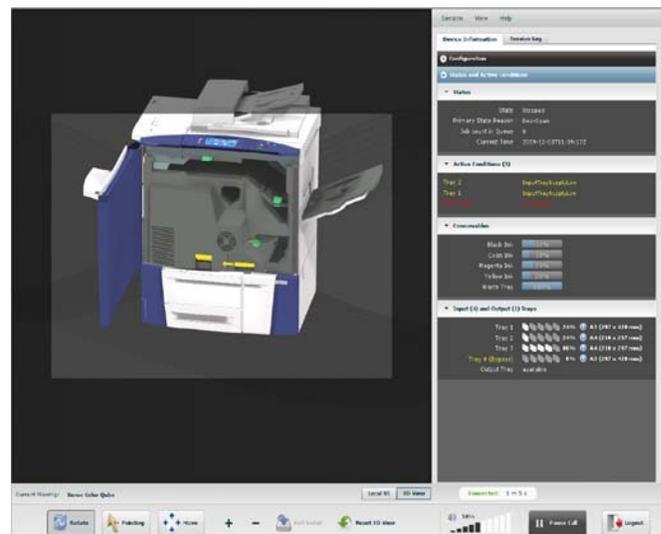


Figure 3. The troubleshooter interface.

The customers' viewpoint is displayed on the device interface and will show one of two views depending on the requirements of the troubleshooting situation. View 1 consists of the 3D representation of the device – with which the customer can interact by indicating device parts to the troubleshooter through the touch screen or by interacting with the physical machine itself. View 2 consists of the LUI with which the customer can interact as normal, or which can be operated by the troubleshooter. In addition to these

viewpoints the customer has controls to adjust the call volume or end the call.

The troubleshooters interface displays more information. Machine state information is shown to the right and the graphical view, onto either 3D representation or LUI, is on the left. Various controls for changing the view or interacting with the display are at the bottom. The troubleshooter's screen is bigger than the customers, so to facilitate reciprocal viewpoints what the customer can see is highlighted (grey area in Figure 3). Troubleshooters have a number of ways of interacting with the 3D representation and through this with the customers. They can view it from different spatial perspectives, to facilitate at-a-glance recognition of problems. They can indicate device parts, e.g. a door, or select an action the user should perform, e.g. removing a toner cartridge and so on.

Whilst the 3D view supports the execution and monitoring of actions on the mechanical parts of the device, the LUI view supports configuration operations that need to be performed on the UI of the device. In the LUI view (Figure 4) the troubleshooter can see exactly what the caller sees and can interact with the display through their computer just as the caller can interact with their touch screen.



Figure 4. The troubleshooters view of Lighthouse displaying the touch screen and status of a remote device.

It acts very similarly to a remote desktop application with the addition of virtual buttons that enable the troubleshooter to activate remotely the hard buttons of the device control panel. Thus the troubleshooter can drive the interaction with the customer or watch while the customer carries out their instructions. In Figure 4, the troubleshooter is showing the caller how to setup the device to print a fax confirmation page.

In many cases the 3D view and the LUI view will be used in combination in order to solve a problem and can be considered as complementary facets of the problem visualization. For example problems requiring the loading of some paper will involve the manipulation of paper trays that can be monitored in the 3D view and configuration of paper types that will happen at the LUI.

Interaction modes supported through the virtual representation of the device

There are three different modes of interaction with the 3D representation to support the various requirements of troubleshooting: synchronous, step-by-step, and simulation. The default mode of interaction proposed to users is to have the two screens synchronized with the current status of the device. For example, if the front door of the device is open this is shown on both users' interfaces (Figures 2 and 3). Using this mode both users can build a common understanding of the problem through a synchronous investigation of the current situation. The troubleshooter can drive the navigation and can zoom, rotate and point. The pointer is shared – the customer can move it by touching the screen. Figures 5 and 6 show how the pointer appears to the other party.

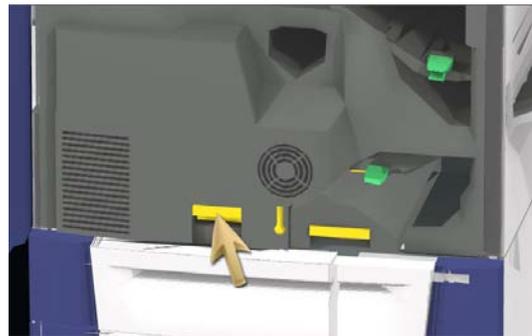


Figure 5. Area pointed by the troubleshooters visible on the customer interface.



Figure 6. Area pointed by the customers visible on the troubleshooters interface.

In the step-by-step mode the troubleshooter can demonstrate how to do particular actions by selecting the part to be operated and choosing the relevant action (Figure 7). This selection displays an animation of the operation to be performed on the customer's interface. Once the operation has been completed, the system returns to the synchronous interaction mode and shows the new status of the device. The troubleshooter will not be able to propose another operation until the current operation is detected as done or s/he decides to abort it.



Figure 7. Troubleshooter can ask to show how to remove the cleaning unit using the contextual menu popping up on top of the cleaning unit 3D model.

Finally, the 3D model can be disconnected from the device status and used as a simulation tool. Thus it can be used as an aide memoire enabling exploration of different aspects of the device. The troubleshooter can switch to the disconnected mode at any time and explore the model independently from the rest of the system status. The 3D representation will be automatically re-synchronized with the actual device status when switching back to the default mode.

How Lighthouse addresses the fieldwork findings

Lighthouse was designed to address the various problematics uncovered in the fieldwork so as to better support the collaboration between the troubleshooter and the customer. With Lighthouse the machine is the infrastructural medium for the troubleshooting support – from the customers’ side they can call support, talk to the troubleshooter and interact around the shared representation through the machine itself. This solves the problem of the telephone not being near the machine. For the troubleshooter, information from the machine sensors can be captured and transmitted. This gives them access to the problem space beyond the customers description enabling them to see the machine status – its physical external (e.g. what doors are open) and internal (e.g. toner levels, etc.) status and its logical status (e.g. LUI settings). This also means that the troubleshooters’ resources for problem visualization have fidelity to the problem machine and as customers perform actions on the machine they can ‘see’ what the customer is doing (since when a customer opens a door, the door on the representation opens). This should enable them to both parcel up their instructions more easily according to customer actions and correct mistakes.

The shared representation enables both parties to indicate parts and so on and the troubleshooters can demonstrate actions. The customer must still translate the instructions from the shared representation to the machine itself but the

visualization of the actions should simplify this work compared to verbal instructions. The shared LUI means that troubleshooters can either get the customer to carry out actions whilst observing them, e.g. for teaching purposes, or can drive the interaction themselves. We aimed to create reciprocal views – in which it was clear what each party could see and therefore enable them to collaborate easily. In the next section we report on the design and findings of a set of user tests to understand whether Lighthouse fulfilled our design hopes.

USER TESTS

Test set-up

We set the tests up to be as realistic as possible given that we were testing a prototype which was not yet fully integrated with either the customers’ or troubleshooters’ machines. The tests were carried out between two sites – a troubleshooting call centre in Canada and the research laboratory in France, which was standing in for the customer site. The tests involved a caller (in France) interacting with a real troubleshooter (in Canada) using Lighthouse. Figure 8 shows the set-up.

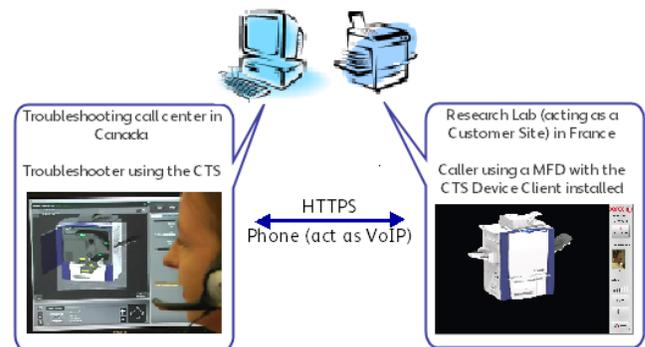


Figure 8. User Test Setup.

The tests involved three experienced troubleshooters, each of whom interacted with two callers. The six callers were recruited locally but were native or fluent English speakers. All worked in an office environment where they used large office printers and multifunction machines as a regular part of their work. Each troubleshooter received a one hour demonstration of Lighthouse in advance and one hour of individual training on the day before the test. Callers did not receive any training, as the application is designed to be accessible to any machine user.

On the callers’ side, the application was installed on a colour MFD. A couple of features were mocked-up for the purposes of the test, because they required engineering work beyond the scope of our research laboratory because they were relying on some integration work that would have required the involvement of the engineering teams of the device, which was too early and costly given the stage of the project. 1) The callers interacted with the application through a touch screen overlaid onto the device’s touch screen as this enabled a Flash version of the Lighthouse

client to be run and this enabled us to manage the integration of the client with the standard UI pages of the MFD. Interactions with the secondary screen were exactly the same as with the integrated touch screen, the only difference being that it also provided the Lighthouse client. 2) A conference phone was used for the voice communication in lieu of the intended VOIP capability. The receiver was attached to the machine. On the troubleshooters' side, troubleshooters used a computer that was configured for the test instead of using their own PC. Further, Lighthouse replaced the troubleshooting application that they typically used. Due to technical issues that arose during set-up, the Lighthouse web client was not hosted on the test computer as intended. Instead it was hosted on a server in France and accessed through Virtual Network Computing (VNC) software. Unfortunately, this added delays to screen updates on the troubleshooters side, the result being a non-synchronized connection with the caller. Both troubleshooters and callers adapted reasonably well but it did cause frustration. The delays would be unacceptable under ordinary circumstances and our assumption is that normally Lighthouse can provide a synchronized audio and visual experience.

In collaboration with a subject-matter expert, six realistic troubleshooting scenarios were developed mindful of the constraints that they a) were possible to mock-up on the MFD and b) required full exploration of Lighthouse – including interaction with the MFD mechanical parts and LUI. Scenarios included paper jams, connectivity issues and administration options. Each caller undertook three scenarios and each troubleshooter encountered all six of them. In each case the caller was asked to contact customer support to make the machine operational. A usability expert was located at both sites to facilitate the test session and two types of data were collected during the test: (1) observations of user behaviour and comments and (2) objective and subjective usability measures administered through during- and post-test questionnaires and interviews.

Test findings

Overall the results of the tests were very positive – 100% of problems were solved and the system was well regarded by both troubleshooters and callers. Troubleshooters especially liked Lighthouse and said that it would help them greatly in their job. They agreed that it was easy to use, improved communication, provided valuable information and made it easier to solve problems more quickly. Callers said it was better than phone-only support and that the interface made it easier to follow instructions. They liked the 3D representation and reported that it was natural that the troubleshooter could access their machine through the system. In the following sections we examine how Lighthouse performed in relation to the identified problematics.

Impact on operators of having support resources which reflect callers machine state

There are three main sets of information the troubleshooters receive from the callers' device: 1) internal machine data, 2) the state of the physical machine and its parts, and 3) the LUI state. At the start of the call the Lighthouse helped troubleshooters more *quickly understand the device's state* and potential issues as they could see fault messages, tray information and so on thus eliminating the need to ask callers to do things like print and read a configuration report. In the post-test interviews, callers described how not having to explain the problem in technical terms to the troubleshooters was a major advantage. It is important that *fidelity to the customers' machine* is maintained however, as was seen when a bug in the system prevented the tray status information from updating during the troubleshooting session. The result in one call was that the user followed the troubleshooter's instruction to load heavyweight paper to the by-pass tray. However, because this change was not updated, the troubleshooter thought there was no paper in the by-pass tray and it took some interaction between caller and troubleshooters to sort out the resulting confusion.

A disadvantage of the system is that is easy for troubleshooters to be overwhelmed with data and consequently miss the most salient information or forget a step in an operation. Because of the way the information is presented troubleshooters and callers may not always attend to the same things. In one session, there were two jams in the machine at once. The LUI showed 'Jam in Paper Transport' and an animation of opening Tray 1 to clear the jam. The troubleshooter focused on that jam. The caller, on the other hand, standing next to the machine, saw that paper was jammed in the By-pass Tray and wanted to clear that. The model, of course is just a representation of the real machine, and breakdowns in fidelity can cause interactional problems. Although data on this jam was available to the troubleshooter it was embedded in a mass of other information and he attended to what was most obvious to him – the jam shown on the LUI, whereas the caller attended to what was most obvious to *him* – the paper visibly stuck in the tray. The caller was quite frustrated, reporting "I didn't want to be told that I was wrong when I could see the paper was jammed in tray 4". Certainly the machine information could be clarified, e.g. by showing machine faults on the 3D representation itself, but the representation can never show everything about machine state and troubleshooters need to use the representation in combination with the callers explanations. Integrating the system smoothly into the interaction between the troubleshooters and user will take time and practice.

Using indicative resources

Lighthouse helped troubleshooters to show, instruct and teach customers via the 3D model and the shared LUI. On the whole they adapted well to using the 3D representation and made use of a variety of the features – pointing, actions on parts and so on. There seemed to be real benefit to using

the representation and customers were in the main quickly able to understand what they should do and then carry out those actions on the device without any major problem. We had anticipated that there might be translation problems between the instructions shown on the 3D model and putting them in place on the machine itself, but actually problems were rare. From the callers' side the only complaint was that troubleshooters overused the more advanced features of the model for simple operations, e.g. pointing first and then demonstrating how to pull out a tray, when once located callers knew *how* to do it.

Another observation was that troubleshooters tended to switch to the 3D view even in cases where there were instructions on the LUI, e.g. for jam clearance, and this at times slowed down the session. For example, troubleshooters moved to the 3D model and showed which door to open, during which time, customers were often itching to open the relevant door but waited politely. One customer actually said "shall I just follow the instructions on screen?". In later sessions at least one troubleshooter used the on-screen instructions. In the LUI view troubleshooters cannot see customers' actions on the machine (and therefore correct or help if required). A possible solution would be to allow the troubleshooters to monitor the 3D model through an inset in the LUI view. Thus, the users can follow the on-screen troubleshooting instructions (e.g., jam clearance) while the troubleshooters watch their interactions with the physical device to ensure that the users are on the right track. In this case, troubleshooters only need to switch to the 3D view to guide the users when necessary.

On the troubleshooters' side, although on the whole they quickly learned to manipulate the model, they had some problems: switching between interaction modes (rotate, point, zoom) requires troubleshooters to move back and forth between the 3D model and the command buttons panel at the bottom of the screen. For example, in order to show users how to pull out the waste tray, troubleshooters selected the Pointer tool at the bottom of the screen and then moved to the 3D model to show where the waste tray is located. They then went back to the bottom of the screen to select another mode (the contextual menus cannot be opened in the Pointer mode) and moved the mouse once again back to the waste tray to issue the open request. This was rather slow. Another problem was difficulty highlighting the machine components in order to bring up the popup menu. This technical problem could be either due to the angle and the zoom level of the 3D model, or because Lighthouse was accessed through the VNC viewer. If it is not an artifact of the test conditions it might be solved by enabling navigation through a mixture of text and the visual components. For example, a list of components could be included and when selected by a troubleshooter the 3D model automatically rotates and zooms to show a good view angle of that component. So 'fax ports' or 'show fax ports' would spin the machine and zoom to the part.

Troubleshooters interacted with customers around the LUI by either driving the interaction themselves or instructing the customer to drive it and following their progress. Customers were happy with both methods. Troubleshooters would like a pointing tool on the LUI to better direct customers. One problem that occurred when using the LUI is that at its edge are some hard buttons which the caller needed to press and callers had real troubles finding these. This is because we had not provided the same reciprocal views onto the LUI, i.e. the troubleshooters saw the hard buttons on their representation of the LUI and tried to point to them, but the callers could not see this since they were next to rather than on the LUI. This trouble tended to be resolved by the troubleshooters operating these buttons themselves.

Seeing customers actions and situating instructions in call flow

Troubleshooters used the 3D representation and the shared LUI to monitor the users' actions on the machine and to situate their instructions according to previous actions and address errors. One caller was acting on the LUI before the troubleshooter had explained how to select paper settings, and pressed confirm twice without changing the paper type. Since the troubleshooter could 'see' her do this he was able to solve the problem and explain how the machine works (detects paper size but not weight and type).

Despite in most cases being smoothly integrated into the interaction the current shared 3D representation does not solve all the interactional problems around machine parts. In one session the troubleshooter asked the user to open Tray 2, instead the user opened the bottom left door and the troubleshooter did not notice and correct this. In another session we saw that customers expect the troubleshooter to be monitoring their actions through the 3D representation: waiting after opening the front door, finally prompting 'ok it's open'. He clearly expected the troubleshooter to time the instructions around his actions. Despite small breakdowns most of the interactions ran smoothly. It is important though that the delay between sites is minimal for the 3D representation to be smoothly integrated into the ongoing interaction. In addition, 1) the 3D representation is not available in all views and 2) troubleshooters have a lot of information to attend to including a knowledge base completely outside of Lighthouse which provides them with problem solutions. Therefore they might not always be attending to the representation. Further investigation would be needed to see if this is simply a matter of learning or if system adjustments could make it easier to focus on the customers actions whilst doing other activities. Certainly an improvement requested by all the troubleshooters was to integrate the knowledge base with the 3D representation and this is something we are already working on.

Of course, the troubleshooters still cannot actually see the user and so cannot see for example that the user has understood and is *waiting* to undertake an action. Certainly from the tests it seems that it is important that the

troubleshooters use the *minimum and fastest set-up* for ensuring customer understanding – for example showing actions for simple steps such as opening doors is rarely necessary. On the callers side even for more complex tasks the time taken to manipulate the model (rotating, zooming, pointing, actions) was often longer than the giving of the verbal instruction. Obviously where it prevents errors this wait is worthwhile and in most cases it wasn't extreme – only in some cases were callers visibly frustrated. However it should be minimized as much as possible. As troubleshooters become familiar with how to work the model and how to incorporate it into the troubleshooting session their interaction with the model should become more fluid. All the same, a key design improvement is to make interaction with the model easier. A better mix of text and visual interactions might help, e.g. enabling labeling of parts (e.g. doors) on a mouse click, leaving the simulations for the more complicated steps – removing parts and so on – or for customers who need that extra bit of help.

In the introduction we discussed the importance of reciprocal views – that is of *knowing what is and is not being communicated to the other side*. For the 3D representation the feature we had put in place to ensure the troubleshooters knew what the caller could see for the most part worked well, however as mentioned breakdowns between the fidelity of the model and the machine did occur and cause interactional problems. The importance of having clear understanding of the others viewpoint was again clearly demonstrated with the LUI where it was not clear to troubleshooters what was visible to the customer. Because for the customer the UI consists of a touch-screen with some hard buttons at the side and for the troubleshooters it is all represented on-screen some confusion arose with, for example, troubleshooters trying to point to the hard buttons. It is important then to make it clear to the troubleshooters what parts of their screen the customer can see, for example using the same grey overlay used for the 3D representation.

DISCUSSION

In this paper we have outlined how a field study of phone technical support for office devices led us to design a system which used a shared virtual representation of the troubleshooting problem to address the uncovered problematics. We chose this shared representation because we believed that it could enable the interacting parties to 1) mutually orientate to the problem – reducing the requirement for technical explanations from the non-technical callers, 2) indicate relevant parts and actions, 3) situate instructions in the ongoing flow of activity and 4) crucially that it would provide reciprocal viewpoints and thus avoid the problems of fractured ecologies which can arise when video is used as the medium for sharing information. At the same time the system does not require expensive additional equipment on the part of the user but it rather relies on existing sensors. We believe that the user tests which we undertook provide a first demonstration of

the utility of this system, as well as highlighting some key places where improvement might be carried out.

It is positive that troubleshooters after just a short training session could largely master Lighthouse, despite its wealth of information, and that both the callers and the troubleshooters thought that it improved the sessions. Most of the interactional issues we reported are minor and occasional and we found that users did not seem to have problems associating the representation with the actual object. This we believe stems from the nature of office devices which have been designed to be user repairable (with large coloured levers, tool-free removal of parts and so on) and it would be interesting to observe such a model in use with more complicated devices such as car engines. Although the tests were largely successful, the representation of the device is just that – a representation – and where fidelity broke down interactional troubles arose. We have suggested some potential solutions to the breakdowns we saw but the representation cannot show everything, so listening to the caller remains key.

Time is critical and enabling the fluid integration of support resources into the interaction is vital. At times the use of the 3D representation seemed too cumbersome for the purposes of the call. Test conditions might have contributed to overuse of the 3D representation as troubleshooters had been asked to use Lighthouse to solve the customers problem. However, some system improvements could make interacting with the 3D model more effective (labelling, rotate and zoom, etc.).

So how does Lighthouse compare to other systems for remote collaboration around physical objects? As with the video-based gesture systems, our system provides a shared workspace view and moreover provides reciprocal viewpoints, which not all of the video systems do. Although our system does not enable naturalistic gestures it has other functionality, for example the ability to demonstrate actions to be undertaken on the machine. Such functionality can be put in place because of the nature of the setting: there are only so many predefined actions which can be undertaken in the normal line of troubleshooting and these can be modelled and incorporated into the representation. They can then be detected by the systems sensors. Clearly the nature of the task has strong implications for the most effective support mechanisms. Superimposed gestures using video intensive systems might be most useful for tasks with a variety of actions not easily pre-defined, which are to be undertaken in a small constrained workspace. However, once the workspace requires navigation by the worker, such systems would become more complicated or costly to implement. Safety-critical situations may enable the implementation of high tech AR solutions. In all these situations the importance of reciprocal views and of enabling mutual orientation and indication remain but there are many possible ways of doing this. Certainly, we believe we have produced a system which is good enough for this setting, requiring minimal extra equipment and that

ethnography was a key factor in doing this. The ethnographic study enabled us to understand the key constraints of the real work environment and by revealing the contingencies of this particular situation and the work within it are, it played a key role in inspiring Lighthouse.

REFERENCES

1. Abt, C. *Serious Games*. (1970), The Viking Press.
2. Baker, C., Emmison, M., Firth, A. Calibrating for competence in calls to technical support. In *Calling for Help*. Baker, C., Emmison, M., Firth, A (eds) (2005), John Benjamins.
3. Bauer, M., Kortuem, G., and Segall, Z. Where Are You Pointing At? A Study of Remote Collaboration in a Wearable Videoconference System. In *Proc. ISWC'99*, IEEE Computer Society (1999), 151—158.
4. Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., and Sommerville, I. Ethnographically informed systems design for air traffic control. In *Proc. CSCW'92*. (1992), 123-129.
5. Castellani, S., Grasso, A., O'Neill, J., and Roulland, F. Designing Technology as an Embedded Resource for Troubleshooting. *Journal of CSCW*, 18 (2-3) (2009), 199--227.
6. Crabtree, A. *Designing Collaborative Systems: A practical guide to ethnography*. (2003), London: Springer-Verlag.
7. Friedrich, W. ARVIKA-Augmented Reality for Development, Production and Services. *ISMAR'02*, IEEE Computer Society (2002), 3-4.
8. Fussell, S. R., Kraut, R. E., and Siegel, J. Coordination of communication: effects of shared visual context on collaborative work. In *CSCW'00*, ACM (2000), 21-30.
9. Gutwin, C. and Penner, R. Improving interpretation of remote gestures with telepointer traces. *CSCW'02*, ACM (2002), 49-57.
10. Heath, C and Luff, P. Disembodied conduct: communication through video in a multimedia office environment. *CHI'91*, ACM Press (1991), 99-103.
11. Heath, C. and Luff, P. Media Spaces and Communicative Asymmetries: Preliminary observations of Video-mediated Interaction. *HCI* 7(3) (1992), 315-346.
12. Henderson, S. and Feiner, S. Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret. *ISMAR '09* (2009), 135-144.
13. Hirzinger, G., Brunner, B., Dietrich, J., and Heindl, J. ROTEX-the first remotely controlled robot in space. *Proc. IEEE International Conference on Robotics and Automation* (1994), 3, 2604-2611.
14. Kirk, D., Rodden, T. and Stanton Fraser, D. Turn It This Way: Grounding Collaborative Action with Remote Gestures. *CHI'07* . ACM (2007), 1039-1048.
15. Kirk, D. S. and Stanton Fraser, D. Comparing Remote Gesture Technologies For Supporting Collaborative Physical Tasks. *CHI'06* ACM (2006), 1191-1200.
16. Kraut, R. E., Miller, M. D., and Siegel, J. Collaboration in performance of physical tasks: Effects on outcomes and communication. *CSCW'96*, ACM (1996), 57-66.
17. Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., and Mitsuishi, M. GestureMan: A mobile robot that embodies a remote instructor's actions. *CSCW 2000*, ACM (2000), 155-162
18. Leroux, C., Guerrand, M., Leroy, C., Méasson, Y., and Boukarri, B. MAGRITTE: A graphic supervisor for remote handling interventions. *ESA Workshop on Advanced Space Technologies for Robotics and Automation* (2004), 471-478.
19. Luff, P., Heath, C., Kuzuoka, H., Hindmarsh, J., Yamazaki, K., and Oyama, S. Fractured Ecologies: Creating Environments for Collaboration. *HCI Special Issue: Talking about things*. 18 (1&2) (2003), 51-84.
20. Milgram, P. and Kishino, F. Taxonomy of Mixed Reality Visual Displays, *IEICE Transactions on Information and Systems*, E77-D (12) (1994), 1321-1329.
21. O'Neill, J., Castellani, S., Grasso, A., Tolmie, P., and Roulland, F. Representations can be good enough. *Proc. ECSCW '05*, Springer (2005), 267--286.
22. O'Neill, J. Making and breaking troubleshooting logics: Diagnosis in office settings. In Buscher, M., Goodwin, D. & Mesman, J. (eds) *Ethnographies of Diagnostic Work*. (2010) Palgrave Macmillan.
23. Ou, J., Fussell, S., Chen, X., Setlock, L., & Yang, J. (2003) Gestural communication over video stream : supporting multimodal interaction for remote collaborative physical tasks. *ICMI'03*. ACM. 242-249
24. Ranjan, B., Birnholtz, J.P., and Balakrishnan, R. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. *CHI'07*. ACM. (2007), 1177-1186.
25. Schmidt, K. and Bannon, L. Taking CSCW Seriously. *Journal of CSCW*. 1 (1-2) (1992), 7-40.
26. Segond, F. and Parmentier, T. NLP serving the cause of language learning. *Proc. eLearning for Computational Linguistics and Computational Linguistics for eLearning Workshop*(2004), ACM, 11-17.
27. See <http://www.americasarmy.com>.
28. See <http://virtualbattlespace.vbs2.com>.
29. See <http://www.tpld.net>.