Designing Technology as an Embedded Resource for Troubleshooting

25 April 08

Stefania Castellani, Antonietta Grasso, Jacki O'Neill, and Frederic Roulland

Email:{Firstname.Lastname}@xrce.xerox.com

Xerox Research Centre Europe

6, Chemin de Maupertuis, 38240 Meylan, France

Tel: +33 4 76 61 50 50, Fax: +33 4 76 61 50 99

Abstract

In this paper we describe a number of technologies which we designed to provide support for customers troubleshooting problems with their office devices. The technologies aim to support both self-conducted and expert-supported troubleshooting and to provide a seamless route from one type of support to another. The designs are grounded in the findings of an ethnographic study of a troubleshooting call centre for office devices. We use the notion of the affordances of different assemblies of people, resources, technologies and spaces to inspire design for the different troubleshooting situations. Through our fieldwork and our technology envisionments we uncovered a number of dislocations between various aspects of the troubleshooting assemblies: 1) a physical dislocation between the site of the problem and the site of problem resolution; 2) a conceptual dislocation between the users' knowledge and the troubleshooting resources and 3) a logical dislocation between the support resources and the ailing device itself. The technology we propose attempts to address these dislocations by embedding the troubleshooting resources in the device itself, thus harmonizing the various

elements and capturing, where possible, the haecceities - the 'just thisness' - of the each particular troubleshooting situation.

Keywords: device troubleshooting, diagnostics, ethnography, ethnomethodology, embedded systems, immediate and remote help-giving, work practice studies, help systems, situated action, socio-technical assemblies.

1 Introduction

Remote device troubleshooting is an interesting area for CSCW: there are different possible configurations of support with different implications for design and thinking about how one might support remote diagnosis involves addressing key issues for CSCW:

- Where there is a remote expert: How do you make the properties of an object available at a remote site? How do you support and enhance actors ability to mutually orient to the ailing object?
- Where there is no remote expert: How do you make the resources needed for diagnosing and repairing the device available to the local actor when, how and as they need them?

Trying to address these issues involves touching on topics at the core of CSCW: the situated nature of action and the affordances of different assemblies of people, materials, technologies and spaces (local, virtual…). By affordances we mean that these different assemblies make available (and visible) certain courses of action while constraining others.

You will find a discussion of the device diagnosis literature at the end of this paper, positioned there so that we can discuss it in the light of the findings and technologies that we describe in this paper.

Our research draws heavily on concepts introduced by Lucy Suchman (1987, 2007). Suchman introduced the term *situated action* in her book, Plans and Situated Actions, as a way to reformulate the problem of purposeful action. "To designate the alternative that ethnomethodology suggests – more a reformulation of the problem of purposeful action, and a research programme, than an accomplished theory – I have introduced the term *situated action*" (Suchman, 1987. 49-50). "The aim of research […] [is] to explore the relation of knowledge and action *to the particular circumstances* in which knowing and acting invariably occur" (Suchman, 1987. 178-179). Of specific interest to us for the design of technology is that the term *situated action* emphasizes the interrelationship between action, the knowledge and reasoning that surrounds that action and the context in which it takes place.

Suchman later expanded this program of research to "include an orientation to capacities for action comprised of specific configurations of persons and things" (Suchman, 2007. 284) that is human-machine *configurations*. She does this to counteract the artificial separation which is often made between the technical/physical and the social, whilst at the same time not denying the nature of people as "those actants who configure material-semiotic networks, however much we may be simultaneously incorporated into and through them." (Suchman, 2007. 270). Importantly for design and studies of use, different assemblies or configurations of people, materials, technology and spaces have different affordances.

The design work presented here, in the domain of device troubleshooting, was inspired by and is grounded in the fieldwork we undertook at a troubleshooting call centre. The call centre dealt with customers calling in with problems with their office devices (e.g. printers). Typically the caller is the user of the device and therefore rarely an expert in the workings of that device. The designs themselves are parts of different configurations or assemblies of people, materials, technology and spaces (in our case physical and virtual/digital spaces) and we aim to present here how the designs encompassed our understanding of how the different configurations would impact on the work to be undertaken within them. The different configurations are designed so that users can, if necessary, progress from self-conducted troubleshooting to collaboration with an expert by taking into account, for each troubleshooting attempt, the interrelation between the troubleshooting actions themselves, the knowledge and reasoning that informs them, the technologies involved and the context in which they are situated. In particular, we aim to design technology which acts as a *resource* (Suchman, 1987) rather than a directive instrument – supporting rather than replacing - the skills and knowledge of the user.

2 The nature of troubleshooting work

In the organisation we studied, there was an emphasis on moving support away from service engineer visits to customer-conducted troubleshooting, e.g. by calling a support centre (expert-supported); with a further move to self-conducted troubleshooting: encouraging customers to try to fix problems themselves before calling. Self-conducted troubleshooting resources included limited on-device instructions and a searchable online Troubleshooting Knowledge Base (TKB). The TKB is available worldwide and contains a repository of problems (cases) and solutions, organised around families of models. The content is semi-structured and the main part of it consists of a list of problems and an associated list of possible solutions, organised with a title and multimedia content describing the solution. The TKB provides a set of categories the user can browse and a keyword search mechanism where users can enter queries describing the particular problem they are facing.

If the users are not able to fix the problem they can telephone a call centre where an expert will guide them through the troubleshooting, or if necessary, call a service engineer to visit the site. Our design remit was to improve both self-conducted and expert-supported remote troubleshooting. At the time we started this research, the TKB was newly introduced and so not in widespread use by customers, although it was being used by the troubleshooters in the call centre. To understand the work of troubleshooting we therefore conducted a three week ethnographic study of the European Call Centre and carried out an in-lab study of the TKB. Deeper discussions of the field study are presented in O'Neill et al. (2005a), O'Neill et al. (2005b), and Crabtree et al. (2006). This created an interesting design situation however as we were not just redesigning support for the assemblies we could observe, i.e. of local and remote actor(s), technologies (ailing device, help system and communication technologies) and knowledge resources. We were also designing for new assemblies: assemblies of local actor(s), technologies (device and help system) and knowledge resources. There has been previous research which has used call centre studies to gather design requirements for online technologies (Martin et al, 1998) and we supplemented the field study findings with data from other sources including various observational user tests of the current customer resources and the prototypes we developed. This gave us additional insights into the reasoning of users when troubleshooting without expert help, albeit in somewhat artificial situations.

2.1 FIELD STUDY FINDINGS

Before moving onto the design we will briefly recall the field study findings. The process of troubleshooting can be glossed as:

- Operators first elicit an initial problem description from customers. This is often partial and the full problem description may only be provided during the course of the interaction. E.g. multiple symptoms will not necessarily be described at once.
- Next operators and customers collaboratively work up the initial description into a fuller description from which they can begin to arrive at possible solutions. Operators may require additional information about the device, which may involve getting the customer to carry out tests, etc.
- Then operators and customers collaboratively troubleshoot the device, with operators giving the customers instructions and customers reporting back on the results of their actions. Although one might consider diagnosis and resolution as separate activities, in office device troubleshooting there is not always a clear distinction between them. Often diagnosis can only be seen to have been achieved in the resolution of the problem. Problems have many possible causes and it is often only through attempting the appropriate solutions that one can determine what that cause is.

The findings we report here emerged from the field studies, where we uncovered a number of places where the customers and troubleshooters had to engage in extra work, above and beyond that involved in troubleshooting the device itself, to make the troubleshooting work. That is they had to engage in articulation work to overcome the affordances of the assemblies in which they are working. For the purposes of design we have classified these as ‘dislocations’, this is of course an abstraction (Dourish, 2007), but we found it useful for ordering the findings so that we could design across different troubleshooting situations. We refer to them as dislocations because in the various alternative troubleshooting situations the actors, the knowledge resources and the support technologies become dislocated from one another in various ways (i.e. they are not in alignment) and need to be ‘worked in’ to the activity of troubleshooting. We believe that for the purposes of innovative technology design it is useful to consider how greater connections between the various actors and artefacts involved in troubleshooting might reduce some of overhead introduced by these dislocations.

We therefore categorise the dislocations, around and through which the customer and the troubleshooter have to work in order to undertake diagnosis under the following headings:

1. *Physical dislocation* between the site of the problem and the site of problem resolution.
2. *Conceptual dislocation* between the users’ knowledge and the troubleshooting resources.
3. *Logical dislocation* between the support resources and the ailing device itself.

We understand that the categories might be considered somewhat arbitrary and different classification schemas are possible (Bowker & Star, 1999) but we found this split was useful[1].

2.1.1 Physical dislocation

A major problem encountered is that there is a dislocation between the site of the problem and the site of the problem resolution. This manifests itself in a number of ways:

[1] It’s important to point out here that the design was undertaken in a multi-disciplinary team, including ethnographers, designers and developers and thus was undertaken with attention to the full richness of the ethnography of which only a small part is presented here.

1. The problem the customer is encountering is not immediately available to technical support. They are not by the device, able to experience and diagnose the problem and at the same time, customers do not have the expertise to diagnose and fix the problems themselves. Thus the troubleshooting activity is a collaborative achievement – customers report their interactions with the device and they work together with the troubleshooters to create, from their individual understandings of this problem here, a shared understanding of the nature of the problem and its possible solutions. Troubleshooters work to uncover and distil information about this particular problem that direct access might provide for them on inspection, since they cannot directly witness it.

2. Troubleshooters need to situate their instructions in the ongoing interaction between themselves, the customer and the device. However, their resources for understanding the customers' interactions with the device are limited to what they can hear and what the customer tells them. In addition, they are describing physical actions to be undertaken on a real device and, in the absence of the device on which the actions are to be performed, they have a number of techniques for embodying the solution. These include miming as they give instructions, using images and menu maps or going to a model of device itself (all the different device models are present in the call centre). In all these cases, although these resources can help the troubleshooter visualise the problem, they involve enacting the actions remotely from the device and then describing them, using only words, to the users. In addition, the representations of the device are just that, they do not show the actual problem device *as it is now* or the customers' position relative to it. Confusion often arises during the provision of instructions, which door? left of what? and so on.

3. To complicate matters further the phone is rarely near the device, requiring to-ing and fro-ing, human chains (e.g. relaying instructions through additional parties) or flat refusal to troubleshoot. Phones tend to reside in private spaces and copiers in public,

and customers are understandably reluctant to use their mobile phones to call support. There is likely to be a similar problem with online support, as PCs also tend to reside in private office spaces.

2.1.2 CONCEPTUAL DISLOCATION

Customers have a range of skills, knowledge, etc. about devices, their problems and so on. The fact that they have contacted support suggests they are unable to rectify the problem on their own. Troubleshooters work according to the exhibited knowledge of the customers to tailor the support to their needs. Customers' initial problem descriptions are often symptomatic, vernacular, partial and wrapped in redundant (for the troubleshooter) information. Problems are multi-faceted and can be described in a number of ways – customers can describe the symptoms, e.g. 'chewing up the paper'; possible causes e.g. 'the rollers are stuck' and so on. In addition, customers often give extra information beyond the basic problem description, such as the situation in which the problem arose. However, the relevancies for the customer and the troubleshooter are distinct and they need to arrive at a mutual understanding. E.g., the customer may need to understand that whether they were copying something from the glass or the document feeder, or the type of paper they were using may matter, whereas the number of pages they'd copied and so on might not.

Troubleshooters also translate the customers' descriptions into the technical terms of both the TKB and the device. They are skilled at reformulating their search terms into alternatives, where they are not effective. The problem descriptions and solutions in the TKB require a certain amount of technical understanding to associate them with the symptoms customers describe and it is the technical knowledge of the troubleshooters about devices, symptoms and problems that enables them to identify the correct path through the TKB with relative ease once they have managed to carry out a successful search. Equally important as translating customers' terms into those of the problem device and TKB is the translation of the TKB content so that it is understandable to the customer.

2.1.3 LOGICAL DISLOCATION

Devices and their problems have a history and context: problems may reoccur; devices may be frequently or rarely used; cleaned or ignored; by a window or a radiator, etc. However, relevant historical and contextual information is not systematically recorded and made available; rather it may only be uncovered in an incidental fashion. For example, it might be occasioned by customers' comments, such as 'the engineer was just out last week'. Troubleshooters do have records of diagnostic calls, however such records are brief and do not include the work service engineers undertake on site. Customers do not necessarily 'know the device' well enough to provide accurate information: the shared nature of such devices has an impact here, with many users information on problems, etc. is likely to be spread throughout the user population. For the troubleshooters, information which might be useful to the diagnosis is frequently not available and at best only partial.

2.2 FIELDWORK CONCLUSIONS

The ethnography emphasised the social and situated nature of the troubleshooting work. This inspired the design of technology with the aim of mitigating these dislocations by reshaping and extending the available resources for both self-conducted and expert-supported troubleshooting. The assemblies of people, resources, technologies and spaces for the different types of troubleshooting will consist in different combinations of:

- Technologies, being various combinations of
    - Broken device
    - Help system(s)
    - Communication technologies
- Knowledge resources
- People/actors
    - The customer(s) – often a non-expert
    - Human expert (remotely located)

- Spaces
    - Local/remote
    - Physical/digital

So from our fieldwork, what can we understand about the affordances of the different assemblies[2]?

- Expert-supported troubleshooting: In the current situation the troubleshooting activity takes place in at least two physical spaces. The virtual space connecting them is via the telephone. Thus much of the richness of the problem situation, including device history, is not revealed to the expert, who also must translate the information resources s/he uses into words to instruct the customer. Where the phone is not near the device we have three physical spaces adding to the dislocation.
- Self-conducted troubleshooting: It is like there will still be two physical spaces at the local site – one with the ailing device, one with the knowledge resources as the PC for the online system is not likely to be co-located with the device. As above, there is a virtual disconnect between the knowledge resources and the device itself. This is likely to be further compounded by a conceptual dislocation between the customer and the knowledge resources, as there is no longer an expert there to translate and interpret.

The aim of the design was to try to enable the actors to get on with the work of diagnosing and repairing a machine rather than grappling with the help systems provided. For self-conducted troubleshooting we aimed to bring critical features of the user-expert troubleshooting interaction into situations where troubleshooting might be undertaken without an expert, while maintaining the tool as a resource and not a prescriptive assistant. The aim here is to address the physical, logical and conceptual dislocations. Where the expert remains the focus is primarily on the physical and logical dislocations, since the troubleshooters

[2] Obviously each interaction episode can be considered as a different configuration, we necessarily have to abstract for the purposes of design

themselves use their knowledge and skills to provide a bridge between the users knowledge and the knowledge required for troubleshooting.

3 Designing embedded troubleshooting tools

Our ethnography and assessment of the tools provided in support of device troubleshooting showed us a fragmented situation, with, at least for the troubleshooters in the call centre, a deal of articulation work to be done to understand the nature of the troubleshooting situation (Crabtree et al., 2006). Currently the main diagnostic support provided to the customer, the TKB and call centre, are both *remote from* and stand *outside of* the problem device. Our technology design therefore attempts to bring together as much as possible, without physical relocation, all the actors and artefacts involved in the troubleshooting activity – namely the troubleshooting resources/expertise (be they human or TKB), the person acting on the device i.e. the customer, and the ailing device itself – to bridge the current dislocations. This is an approach similar to that taken by for example Buscher et al (2003) in ubiquitous computing: an attempt to make “available and ‘inhabitable’ as much as possible of the configurations of reference materials and features, people’s orientation, gestures, pointing and talk in relation to them.” (139). Our aim is to reconfigure the assemblies or create new assemblies that address the dislocations which were both found in our field sites and arise from our envisionments of the impact our designs might have as we move from one assembly to another.

We have developed a set of integrated mechanisms and systems aimed at harmonizing the various elements and at capturing, where possible, the haecceities - the ‘just thisness’ - of the each particular troubleshooting situation (Garfinkel and Wieder, 1992) and transmitting this to the troubleshooting resources – the call centre operator or the TKB – to be used to aid diagnosis and repair. This work was enabled by a new generation of multi-function devices which have colour touch screens - supporting a higher degree of interactivity – and are networked and increasingly provided with software extensibility capabilities in order to integrate with custom external services. In our case this extensibility was offered by the embedded UI of the device being implemented as a web browser that can access external web services. This provided us with the technical infrastructure necessary to embed the troubleshooting resources in the device itself. The objective of our design has been to reduce the logical, physical and conceptual gaps among the various support tools, moving from a fragmented set of resources to an integrated one, embedded within the device itself, as shown in a schematic way in Figure 1.

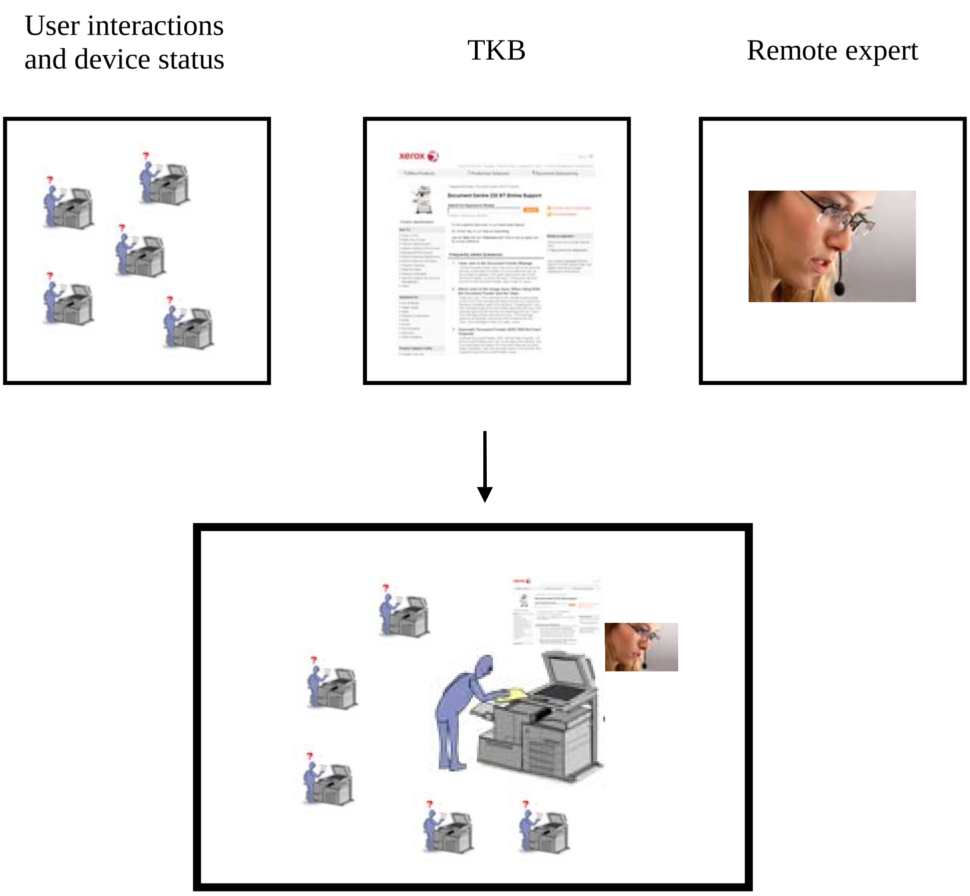

Figure 1: From fragmented to integrated resources.

Due to resources, we have taken a staged approach to development and the designs described below are at various stages of implementation: all the designs have been patented; some have reached fully working prototype stage and are being developed for transfer; others are currently in the early stages of development and are likely to change as they progress; and one, the use of immediate device history, remains as a proof of concept.

3.1 CLOSING THE PHYSICAL GAP

Our first step was to consider how one might reduce the physical dislocation between the broken device and the troubleshooting resources. Closing it could reduce the articulation work necessary to provide a diagnosis and provide an enabler, as we will see later, to closing the logical gap. The physical dislocation can be addressed in two settings – the users' self-support and the expert-support situation.

3.1.1 SELF-CONDUCTED TROUBLESHOOTING

For self-support we wanted to bring the troubleshooting activity into a single physical (and virtual – see section 3.4.2) space: that of the device to be repaired. This led us to design additional *on-device support* by providing access to the TKB through the ailing device providing access to expanded support resources. Devices already provide some limited troubleshooting support, typically suggesting operations to be performed or providing warnings, when a problem that can be detected by the device arises. These instructions lead the user through some simple troubleshooting steps and have the advantage of being activated by a known device state, so do not require search. Presented instructions are triggered by user actions, e.g. opening doors, enabling them to be situated in ongoing user activity. However, this support only covers situations where a one-to-one mapping between a sensed device event and a solution is assumed. It is therefore limited to very few problems and even there it could be biased by a wrong assumption.

We used the embedded UI on the device to act as a web browser and to display on the device itself a version of the online TKB redesigned for the new interaction parameters. This required adapting the pages for the new (smaller) screen dimensions, for the touch-screen interaction and for interaction at the device. For example, query typing requires a soft keyboard displayed on the screen and when combined with smaller screen size, resulted in a design where query-entry and results-viewing were on separate pages. Typing on a touch screen keyboard tends to be slower for most users since it tends to be more of a single-finger task than on a familiar conventional keyboard. In addition interaction at the device is standing, at a public device, reducing further the time likely to be devoted to the diagnostic work (Martin and O'Neill, 1999). These factors and more meant that the interface and interaction mechanisms for the on-device display of the TKB pages were extensively redesigned.

Putting the resources on the device constitutes conceptually a big step forward in reducing the physical dislocation between the device and the troubleshooting resources. However, this new assembly introduces new expectations around 'what the system knows'. Situating the resource on the device creates an expectation (affordance) that that resource is now integral to that device and thus will know the state of the device that was not there with the online system. We took account of this in the design of the system, as will be described in Section 3.3, and it was confirmed during the user testing of the various prototypes. For example, as well as expecting that the system would know its own state (what doors were open and so on), users expected that the suggested results from their search would only include results relevant to the particular configuration of the particular device they were troubleshooting. Thus by physically situating the resources on the device, users expected that these resources would be logically connected. This presented us with an interesting design challenge since the support resources are actually located on a web server remote from the device – they only *appear* to be situated on the device itself – thus passing knowledge between the device controller and the web portal is not a simple matter (see Section 3.3).

Bridging between the device and the troubleshooting resources is a required condition to closing of the logical gap, necessary to enable the problems' context to be traced.

3.1.2 EXPERT-SUPPORTED TROUBLESHOOTING

For expert-supported troubleshooting we wanted to reduce the dislocation by: 1) on the customers side, bringing the troubleshooting into one physical location and 2) enhancing the virtual space between the customer and the remote troubleshooter. To address the first of these we enable the customer to call the call centre directly *through* the device itself, via an on-screen button, using VOIP and a provided headset. To address the second of these we use a secure data and audio end-to-end connection to the support centre to transfer data about the device to the troubleshooter. Data might include the device serial number, sensor information on current device state, historical information, etc. This data is then used by the remote server, in combination with its own stored data for that device - e.g. records of other troubleshooting sessions - to build an initial representation of the problem. This includes a visualisation of the device in its current state which is on both the troubleshooters' and the customers' interfaces. The troubleshooters' representation has additional functionality such as: the means to view this representation from different spatial perspectives to facilitate at-a-glance recognition of problems; control buttons, through which they can interact with the customer, e.g. to indicate a part of the device, select an action the user should perform and so on, by using visual indicators, e.g. LED lights. When the customer performs actions on the device, they are shown on the shared representation, thus the operator can infer, for example, if the customer is following his instructions correctly.

Figure 2 shows the architecture of the collaborative troubleshooting system.

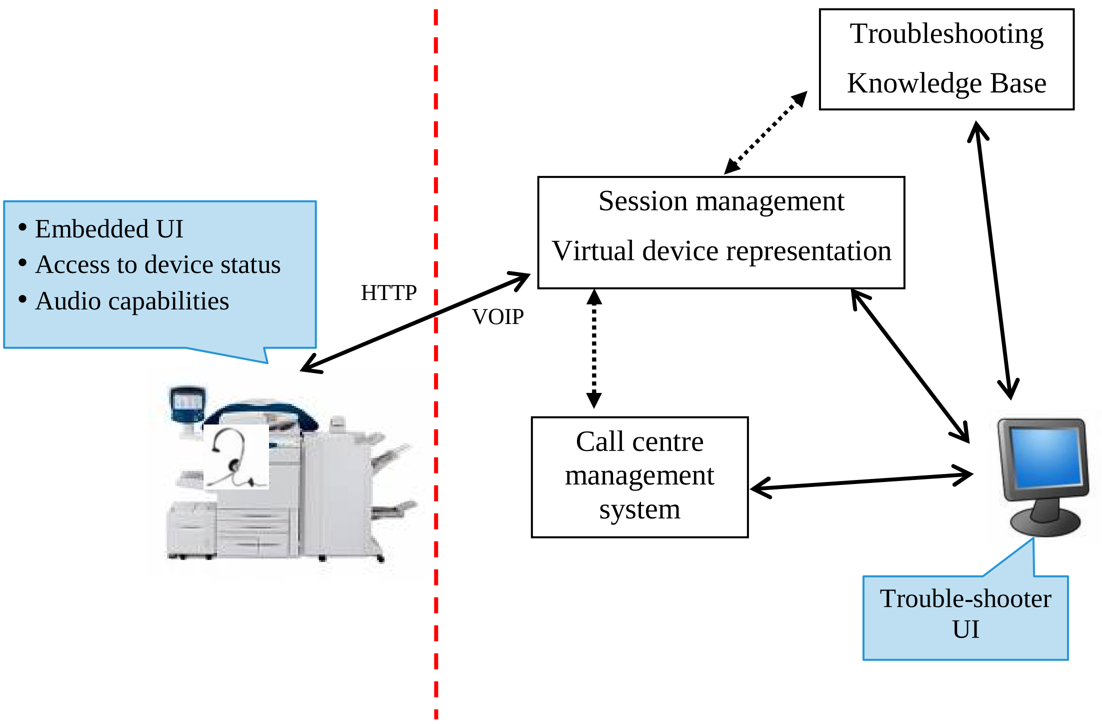

Figure 2: An architecture for collaborative troubleshooting.

The collaborative troubleshooting system was designed to help troubleshooters to understand the current problem state and to situate their instructions in the ongoing stream of activity as the remote troubleshooter would be able to 'see' what the customer had done more or less as it happened and thus give the next instruction. Reciprocal viewpoints are supported and operators and customers will be able to co-ordinate and co-orient around the representation of the object. Although at first it may seem to be a relatively basic and simple representation, this seemingly shallow representation is actually able to capture salient indexical information, unlike troubleshooters existing representations which are unconnected to the ailing device. This means that the haecceities, the 'just thisness' of the problem can be explored and revealed (O'Neill et al, 2005).

3.2 CLOSING THE CONCEPTUAL GAP

For the self-conducted troubleshooting, we predicted - on the basis of the translation work that the troubleshooters do between the customer and the TKB - that there would be a conceptual dislocation between the knowing about the problems of the device (as experienced here and now) and the level of expertise required to successfully search and identify results in the TKB. We therefore wanted to reduce this dislocation between the customer and the knowledge resources. Whilst instructions can be augmented by visual description and user-friendly terminology, traditional search mechanisms offer little support to users. In the user tests, users had real problems expressing their problems in a way that produced good results and identifying right (and wrong) paths through the returned results. To this end we have developed several techniques to help the user. These include a query suggestion mechanism, discussed below and a tree-based refinement mechanism to help users explore the problem space (Roulland et al., 2007). Whereas the refinement mechanism has been tested on the web-based system and seemed to provide some exploratory support, we have not found it to be so useful on the on-device system primarily as users, saw but did not investigate the refinement option.

The query suggestion mechanism was developed to expand the users' queries as they type, according to the contents of the TKB and their synonyms. Suggestions are expressions of TKB terminology that are collected through the natural language processing (NLP) of the TKB content. They help users to express their problems in ways compatible with the TKB terminology even when they have a limited knowledge of the device. These suggestions are ranked according to their importance in the TKB but also according to their popularity on previous searches. We analyze the logs of previous sessions and collect frequencies of the expressions in previous queries taking into consideration whether the expression was used in queries leading to satisfactory results. This ranking strategy is a way to increase the probability that the proposed suggestions will fit the user problem. The first role of this feature is to improve the terminology of the user's queries; however, it also facilitates the query typing when it is performed on the device. Figure 3 is a screen shot where this feature is integrated with the soft keyboard panel in the device UI. In this example we can see how after having typed only the two characters 'c' and 'a' the system suggests a list of TKB expressions around the term 'cartridge' that are the most probable matches of the user's problem according to the previous logs. In addition the different formulations 'install cartridge' and so on give the user indications of the TKB terminology.

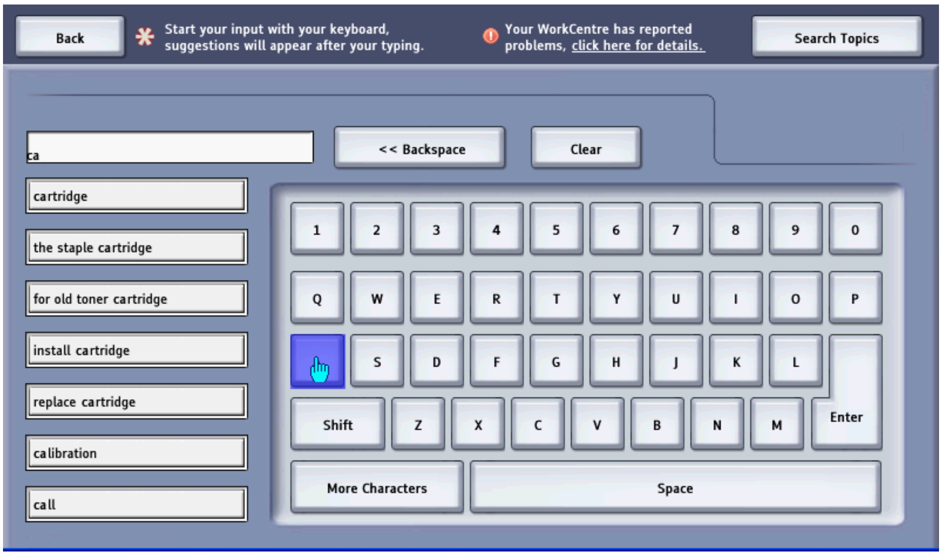

Figure 3: The query suggestion mechanism in the device UI.

3.3 CLOSING THE LOGICAL GAP

In this section, we address the logical dislocation, with a focus on: 1) the history of device usage and 2) using device context to inform troubleshooting. In both these cases we make use of the affordances of the ailing device to sense actions and interactions to: 1) create a history and 2) inform the search of the TKB by making this information available to it.

3.3.1 DETECTING AND USING DEVICE HISTORY

Printers do not currently leverage knowledge about past interactions with the device. So, for example, when a problem occurs and a number of users follow the instructions provided by the device but they are not able to solve the problem, successive other users may each try to solve the problem without knowing of the prior failed attempts. In addition, when a remote expert becomes involved, they often ask the customers to repeat actions they have already undertaken and this can cause frustration on the customers' part. In order to help users to avoid trying out the same unsuccessful sequence of troubleshooting actions, we have developed a method which uses knowledge about past interactions with the device. Our proposal involves the recording of all user interactions with the device in particular those linked to troubleshooting and maintenance operations. Interaction data includes: single actions, sequences of actions and troubleshooting sessions (from problem detection to resolution). Information can be extracted from these records and compared to ideal sequences of actions from the TKB to understand: (1) if some users have somehow (e.g. only partially) followed the troubleshooting instructions given by the device but not fixed the problem and (2) if some of them have tried recurrent sequences of actions even if not suggested as a troubleshooting procedure.

Of course there are a number of approximations and sources of noise in the user interaction data used in this process since not all human actions and faults are detectable or actions may be performed out of sequence. We therefore use the *sequence kernel*, frequently also referred to as *string kernel* (Cancedda et al. 2003, Lodhi et al. 2001), to match ideal sequences with real sequences taking account of the noise. The resulting information could then be provided to the user or the troubleshooter in a number of ways, for example, X users have tried Y action, or it can be used to inform which solutions are shown to the user, e.g. if doing Y doesn't solve the problem, try Z, etc.

3.3.2 USING DEVICE CONTEXT TO INFORM TROUBLESHOOTING

We are primarily thinking about self-conducted troubleshooting here, however, since the troubleshooters also use the TKB, this information might also be used by them. When a user searches for a solution to a problem, the TKB provides a set of results that are independent from the actual configuration and state of the device, whereas the user is interested in fixing this problem here, now, with this device. In order to contextualise troubleshooters searches to the users' particular device, we have designed, and in part implemented, a method of injecting device history, status and diagnostic data into the search (Castellani et al. 2007).

When a user provides a textual description of the problem the system generates a set of possible matches in the TKB and prioritises them according to data gathered from the device. More precisely, the system integrates the following kind of information into the search results:

- Diagnostic data
- Configuration settings
- Device history and current status

We assume that the diagnostic data is provided by a diagnostic engine. When it is possible to establish a diagnosis it takes the form of a list of references to TKB cases/solutions. The diagnostic data is used to increase the visibility of the cases/solutions it references.

Information on device configuration refers to the hardware and software components present in the device. Each piece of configuration is associated with a concept node of the TKB model. This information provides a description, i.e. the validity of each of the concepts linked with a configuration datum, e.g. presence of a finisher. The configuration datum is used to perform a sort of 'negative' inference. When a feature is not present in a device, all the problems classified as relevant when this feature is present get lower visibility, whereas if a feature is present then the problems categorized as relevant when this feature is off get lower visibility. For example, if the model can have one of several types of finisher and that specific device has a finisher of type "professional", solutions which mention other finishers should be less visible.

Moreover, information from recent history and current status of the device could be used to give more context to the presentation of the solutions. In particular, information on manipulations of mechanical parts, e.g. opened trays; functions recently used, e.g. printing from document glass; and recent maintenance operations, e.g. cartridge changed one hour ago, can be used to prioritise solutions or grey out the text describing the actions "already performed" in solution descriptions.

This solution goes some way to addressing the logical dislocation which arises from closing the physical gap and putting the troubleshooting resources on the device itself, by embedding 'knowledge' of the problem device and its current and past states in the troubleshooting resources.

4 Discussion

In this paper we have described the design of technologies for supporting a chain of remote diagnosis, moving from customers self-troubleshooting problems to customers working with a remote expert to solve the problems. We have described the different technological features individually, showing how they were designed to address the different dislocations uncovered in our analysis of the fieldwork and the envisionments of the different assemblies. The aim is to have an integrated system enabling customers to move smoothly from one assembly to another – from device suggested solutions, to the on-device knowledge base, to expert-supported troubleshooting. This itself addresses a what might be seen as a final dislocation – of the different forms of support resources from one another. Conceptually we addressed this by using the ailing device as the infrastructural medium for that troubleshooting[3]. Interestingly, Suchman characterised one crucial asymmetry between humans and devices as being the machines access to the activities of the user only when those activities change its state, whereas people have a much wider awareness of what is going on. This is still, of course, the case in our work but by tying the help resources with the ailing device we aim to make the systems aware of each other and of course where the human expert is involved, they can make inferences from the actions they 'see' changing the device state. This aims to address both the physical dislocation – making the resources available where, when and to whom they are needed, but also by making them somehow situationally-aware also begins to bridge the logical divide.

In terms of other research on device troubleshooting, we have not found other equivalent efforts to make troubleshooting support an integrated service. On the other hand, there are many systems focusing on some of the individual technological features we have described and we illustrate some of them in this section.

There are some methods to enhance human-customer troubleshooting, e.g. in Edmunds et al. (2003) the customer can establish from the printer a phone/data line connection with a troubleshooter and use voice communication and remote diagnostics. "Session data logs" can be used to share data read from the device and inform the troubleshooter. This work is close to our approach for collaborative troubleshooting; however despite enabling initial diagnostic data and device state to be sent to the specialist, the primary interaction channel is still the phone. Troubleshooting therefore still suffers from many of the problems related to the audio channel, such as verbalising instructions, directing users through space, describing device parts using non-technical language, lack of an indexical representation, etc.

[3] We are well aware that this will only be possible in the cases where the parts of the device that are broken do not interfere with the parts of the device being used to fix it. We believe these cases are fewer and in these cases, customers would need to resort to original web-based and telephone support.

Rollins (2003) aims at providing an identical representation of the status of a computer system on the troubleshooter's side by means of a software component on the user's computer able to collect and transmit the status of the system to a remote computer. The approach we have taken aims to represent the status of the system and at the same time provide a framework in which the troubleshooter and user can interact, coordinated by the shared representation of the problem and the actions to be performed. Unlike our approach, this method does not use the system status as a coordination point to perform actions.

Friedrich and W. Wohlgemuth (2002) describe an augmented reality system allowing a mobile on-site user to be instructed or to access documentation via an augmented reality headset, in order to carry out device maintenance. This method provides help according to situational factors (e.g. the users position), but it does not encapsulates information about the current status of the device. Also, it uses a mediating headset to get information whereas the approach we adopted provides the support on the device without requiring users to manipulate expensive and potentially weighty AR headsets.

In the same area of instructing the user from a remote location, there is work on pointing, defining actions, etc. Bauer et al. (1999) describes a reality augmented telepointer for supporting mobile, non-expert fieldworkers undertaking technical activities with the aid of remote experts. Non-experts wear a Head-Mounted Display (HMD) with camera, audio and wireless communication link. The video captures what the non-expert sees and relays it to a PC-based system. The expert is able to view the video information and guide the non-expert's activities through audio instruction and an overlaid ‘pointer’ which the non-expert can see in their HMD. The expert can also freeze the video image to get around difficulties of head movement when trying to point. This is close to our scenario of use and offers similar ostensive capability but is video-based. The frozen object on the PC and HMD has similarities to our notion of a shared representation. However, as with the AR above this is device independent and requires an HMD, as does Nomad Expert Technician System (Microvision) which uses an HMD to project images of information (textual or graphic - generated by PC) onto the technician's eyeball. The technician is able to set the same focal depth for the information as the space being worked upon, allowing for information to be apparently overlaid on the object. This is not a shared and dynamically updated representation as such but rather any means of having rapid, in situ information access. The main difference is that this methodology is not used to facilitate guidance by an actual remote expert.

With respect to the support for searching troubleshooting solutions, two main approaches can be cited: “knowledge-rich” approaches that make use of domain knowledge in the form of decision trees or models of the device, and “knowledge-poor” approaches based on information retrieval (IR) techniques that don't use domain knowledge (with the possible exception of linguistic resources such as term lists or thesauri). The solution we have designed for textual search of troubleshooting instructions falls into the latter class.

Knowledge-rich approaches adapt various techniques from AI to the problem of identifying a solution to a problem: model-based systems, e.g. Peischl and Wotowa (2003), case-based reasoning systems, e.g. Jensen et al. (2001), and rule-based systems, e.g. Aha et al. (1998). One drawback of many knowledge-rich systems is that users who already have a good idea of what they are looking for often find them too restrictive. One system which addresses this problem, by adding keyword search to a case-based reasoning system, is MCRDR (Kang et al. 1997). In the solution we propose, both the refinement tree and the top-ranked results are available at any time. If the user's query is sufficiently precise for the desired solution to appear near the top of the results, then the user can go directly to it.

In IR-based troubleshooting systems, e.g. Eureka (Bobrow and Whalen, 2002), the standard keyword search paradigm is applied to searching in a collection of solution descriptions. This type of interface is very familiar to most users, and can be effective for someone who is familiar with the contents of the TKB but does not meet the needs of a non-technical user. With both approaches the logical gap between the device and the help system remains. With the knowledge-rich approach the conceptual gap between the users understanding of the problem and the help resources is likely to be reduced for *non-expert users* but at the cost of a perceived lack of flexibility. With the IR approach the conceptual gap is likely to remain. Some systems for device problems diagnostics also consider some forms of integration of textual search with device data, e.g. in Lao et al. (2004) for troubleshooting registry configuration issues in PC applications. In their approach diagnostics is based on configuration parameter observations automatically mixed with data from a past problems database. However, in our approach, informed by the ethnography, selected technical data are injected in user searches whilst at the same time leaving the user in control of his search, i.e. support rather than automation. This is an attempt to reduce the logical gap with the device but again the conceptual gap with the user is not addressed.

5 Conclusion

We conducted an ethnographic field study of a troubleshooting call centre which was used to inspire the design of a number of technological solutions covering self-conducted and expert-supported troubleshooting. Unlike other technical attempts that have focussed on enhancing specific troubleshooting features in isolation, we have considered a wide set of resources to be embedded in both the ailing device and the social troubleshooting setting. These technologies were designed to reduce the three types of dislocation which emerged from the field study and the envisionments of the impact of the different assemblies of people, resources, technology and spaces. The interrelationship between troubleshooting actions¸ the knowledge and reasoning that surrounds those actions and the context in which they take place - which sits at the heart of the concept of situated action - have been our central concern. By closing the various types of gaps among the support resources we have proposed a system that reduces the separation between the troubleshooting resources and the troubleshooting situation

6 Acknowledgements

We thank A. Kaplan, C. Roux, K. Pettersson, J. Selin and V. Ciriza for their contributions to the design of refinement mechanisms and the on-the-box interaction; and N. Cancedda for his contribution to the method for avoiding repeated troubleshooting interactions on devices.

7 References

Aha, David W., Tucker Maney, and Len Breslow (1998): Supporting Dialogue Inferencing in Conversational Case-Based Reasoning. In B. Smyth and P. Cunningham (eds): *EWCBR'98. Proceedings of Advances in Case-Based Reasoning, 4th European Workshop, Dublin, Ireland, 1998*. LNCS 1488, Springer, pp. 262--273.

Bowker, G. & Star, S. L. (1999): *Sorting Things Out: Classification and Its Consequences*. Cambridge MA: MIT Press.

Bauer, Martin, Gerd Kortuem, and Zary Segall (1999): "Where Are You Pointing At?" A Study of Remote Collaboration in a Wearable Videoconference System. *Proceedings of the 3rd International Symposium of Wearable Computers (ISWC'99), San Francisco, California, 18-19 October 1999*. IEEE Computer Society, pp. 151--158.

Bobrow, Daniel G. and Jack Whalen (2002): Community knowledge sharing in practice: the Eureka story. *Journal of the Society for Organizational Learning*, Vol. 4, Issue 2.

Buscher, Monika, Kramp, Gunnar & Krogh, Peter Gall (2003) In formation: support for flexibility, mobility, collaboration, and coherence. *Pers Ubiquit Comput* 7. pp. 136-146.

Cancedda, Nicola, Eric Gaussier, Cyril Goutte Cyril and Jean-Michel Renders (2003): Word-Sequence Kernels. *Journal of Machine Learning Research*, 3(February):1059--1082.

Castellani, Stefania, Jacki O'Neill, Frederic Roulland, and Antonietta Grasso (2007): Informing troubleshooting search sessions with device data. *ECSCW Workshop on "CSCW, Technology and Diagnostic Work", Limerick, Ireland, September 24-28*.

Crabtree, Andy, Jacki O'Neill, Peter Tolmie, Stefania Castellani, Tommaso Colombino, and Antonietta Grasso (2006): The practical indispensability of articulation work to immediate and remote help-giving. In P. Hinds and D. Martin (eds): *CSCW'06. Proceedings of the 2006 Conference on Computer Supported Cooperative Work, Banff, Alberta Canada, November 4-8.* ACM New York, pp. 219--228.

Dourish, Paul (2007): Seeing Like an Interface. *Proceedings of OzCHI 2007*. pp.1-8.

Edmunds, C. W., D. Auman, K. R. Mathers, C. S. Lippolis, A. M. Lorenzo and C.-L. Goldstein (2003): Simultaneous voice and data communication for diagnostic procedures in a printing or copying machine. *US-Patent*, 6,665,085.

Friedrich, Wolfgang and Wolfgang Wohlgemuth (2002): System for, and method of, situation-relevant assistance to interaction with the aid of augmented-reality technologies, US-Patent-Application 2002/0046368A1.

Garfinkel, H. and D. L. Wieder (1992): Two incommensurable, asymmetrically alternate, technologies of social analysis. In G. Watson and R. M. Seiler (eds.), *Talk in context: contributions to ethnomethodology*, 1992. New York: Sage, 175--206.

Kang, Byeong Ho, Kenichi Yoshida, Hiroshi Motoda, and Paul Compton (1997): Help Desk System with Intelligent Interface. *Applied Artificial Intelligence*, Vol. 11, Num. 7, 1 December, pp. 611--631(21).

Jensen, Finn V., Claus Skaanning, and Uffe Kjaerulff (2001): The SACSO system for Troubleshooting of Printing Systems. In H. H. Lund, B. H. Mayoh and J. W. Perram (eds): *SCAI'01. Proceedings of the 7th Scandinavian Conference on Artificial Intelligence, Odense,* *Denmark, 19-21 February, 2001*. Frontiers in Artificial Intelligence and Applications, pp. 67--79.

Lao, Ni, Ji-Rong Wen, Wei-Ying Ma, Yi-Min Wang (2004): Combining High Level Symptom Descriptions and Low Level State Information for Configuration Fault Diagnosis. *Proceedings of the 18th Conference on Systems Administration (LISA 2004), Atlanta, USA, November 14-19, 2004*. USENIX, pp. 151--158.

Lodhi, Huma, Nello Cristianini, John Shawe-Taylor and Christopher Watkins (2001): Text Classification Using String Kernels. *Advances in Neural Information Processing Systems* 13. MIT Press, pp. 563--569.

Martin, David, Wastell, David & Bowers, John. (1998) Ethnographically Informed Systems Design (EISD): The development of an Internet-based electronic banking application. *Proceedings of European Conference on Information Systems (ECIS)*, June 1998.

Martin, David, Dave Wastell, Jacki O'Neill and Richard Varey (1999): Pushing All The Right Buttons: Lessons from two case studies of touch-screen kiosks. In M. A. Sasse and C. Johnson (eds): *Interact'99. Proceedings of Interact, Edinburgh, Scotland, August 30 to September 3, 1999*. IOS Press.

Microvision: Microvision Inc.'s Nomad Expert Technician System**,** **"**http://www.mvis.com/nomadexpert/info.html", and "http://www.mvis.com/pdfs/nomad_industry.wp.pdf".

O'Neill, Jacki, Stefania Castellani, Antonietta Grasso, Peter Tolmie, and Frederic Roulland (2005a): Representations can be good enough. In H. Gellersen, K. Schmidt, M. Beaudouin-Lafon and W. Mackay (eds): *ECSCW'05*. *Proceedings of 9th European Conference on Computer Supported Cooperative Work, Paris, France, September 18-22, 2005*. Springer, pp. 267--286.

O'Neill, Jacki, Stefania Castellani, Antonietta Grasso, and Peter Tolmie (2005b): Using Real-Life Troubleshooting Interactions to Inform Self-Assistance Design. In M. F. Costabile and F. Paterno' (eds): *Interact'05. Proceedings of Interact, Rome, Italy, September 12-16, 2005*. LNCS 3585, Springer, pp. 377--390.

Peischl B. and F. Wotowa (2003): "Model-based diagnosis or reasoning from first principles". *IEEE Intelligent Systems*, Vol. 18, Num. 3, 2003, pp. 32-37.

Rollins, D. L. (2003): Graphical representation of system information on a remote computer. *US-Patent*, 6,664,988.

Roulland, Frederic, Aaron Kaplan, Stefania Castellani, Antonietta Grasso, Claude Roux, Jacki O'Neill, and Karin Pettersson (2007): Query Reformulation and Refinement Using NLP-Based Sentence Clustering. In G. Amati, C. Carpineto and G. Romano (eds): *ECIR'07. Proceedings of 29th European Conference on Information Retrieval, Rome, Italy, April 2-5*. LNCS 4425, Springer.

Suchman, Lucy (1987): *Plans and Situated Action: The Problem of the Human-Machine Communication*. Cambridge: Cambridge University Press.

Suchman, Lucy (2007) *Human-Machine Reconfigurations: Plans and Situated Actions*. 2nd Ed. Cambridge: Cambridge University Press.