Working paper : Diagnosing machine problems on the phone: technological inspirations from an ethnography of user-expert interaction

J. O'Neill, S. Castellani, A. Grasso and F. Roulland

Xerox Research Centre Europe

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

Not to be distributed beyond workshop participants

INTRODUCTION

This paper describes an ethnography of a call centre where troubleshooters attempt to diagnose and fix problems with large office devices, e.g. printers over the phone. The help-seeker is generally the user of the device, at the site the device is installed, and often non-expert in the technicalities of the device. Diagnosis and repair is a collaborative activity which consists of a number of activities. These include: 1) the collaborative *working up* of a problem description from customers' initial reports; 2) *translation* from customers terminology to that of the device/troubleshooting resources and from the technical terminology of machine and knowledge-base to the customers language; 3) *mediation* - troubleshooters mediate between technology resources and customers and customers mediate between machines and troubleshooters. The distributed nature of the activity means that both troubleshooter and customer engage in co-ordination work to make this mediation work. The troubleshooter has access to knowledge whereas the customer has access to machine. Diagnosis and repair usually involves physical manipulations of machine or its software from instructions over the phone. The troubleshooter only has limited access to what the customer is doing and must verbalise all instructions. In addition there is often a dislocation between the site of the problem and the site of problem resolution, that is, telephones are rarely by shared office devices.

The ethnography emphasised the social nature of troubleshooting work and this understanding led to some inspirations for technology design to support such work along different dimensions: online support, collaborative troubleshooting, and between the customer and the machine. In this paper we first describe the case study findings and then we illustrate the technology ideas that they have inspired. The aim of the technology design was to bring critical features of the user-expert troubleshooting interaction into situations where troubleshooting might be undertaken without an expert; and where the expert remains to better support their work. Our wider aim is to enhance the design of the technical by basing it clearly in the social.

ETHNOGRAPHY OF USER-EXPERT INTERACTION

We conducted a three week ethnographic study of a European Call Centre for a large copier and office device company. The call centre provides telephone support to locations across Europe for customers with problems with their office devices (copiers, printers, etc.). Operators lead the customer through a process of troubleshooting the problematic device over the phone. If the problem cannot be solved by the troubleshooter, they can pass the customer onto a second level of technical support or place a call out to a service engineer who will visit the site. Some problems, such as broken parts, cannot be solved over the phone so an immediate call out is made. During the three weeks a number of troubleshooters were observed carrying out their work. Field data was collected through field notes, video and audio recordings[1]. The observed work of troubleshooting could be summarised as:

1. Operators first elicit an initial problem description from the customers. This initial problem description is often partial and the full description of the problem, as it appears to the customer, may be provided during the course of the interaction. For instance multiple symptoms will not necessarily be described all at once.
2. Next operators and customers collaboratively work up the initial problem report into a fuller description from which they can begin to arrive at possible solutions. The operators may require additional information about the machine and this may involve getting the customer to carry out tests on the machine, provide readings and so on.
3. Then the operators and customers collaboratively troubleshoot the machine, with operators giving the customers instructions to carry out and customers reporting back on the results of their actions.

THE WORK OF TROUBLESHOOTING

The problem the customer is encountering is not immediately available to technical support. They are not by the machine, able to experience and diagnose the problem. At the same time, customers do not have the expertise to diagnose and fix the problems themselves, having called the helpline. Therefore the customers and troubleshooters collaboratively work up a diagnosis; customers report their interactions with the machine and both the customers and troubleshooters bring to the interaction their understandings of the possible machine problem. They work together to create a shared understanding of the nature of the problem and its possible solutions. Troubleshooters have a number of resources to hand, beyond their own knowledge, to help them work towards diagnosis and solution.

The first stage of the interaction is the working up of a problem description. Customers do not simply describe problems in their entirety, rather their initial 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 they see, e.g. ‘chewing up the paper’; what they think the cause of the problem is, e.g. ‘the rollers are stuck’ and so on. In addition customers often give extra information beyond the basic problem description, such as about 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, at least for this activity. Thus the customer may need to be led to understand that whether they were copying something from the glass or the document feeder may matter, whilst the troubleshooter may not be much concerned to probe the variety of other devices the customer has also had go wrong that morning. However, although not directly relevant for diagnosis such information informs the troubleshooting interaction, as troubleshooters need to manage the interaction, particularly where customers are frustrated or angry. Since troubleshooters are not able to directly witness the behaviour of the device, they have to distil and configure information that direct access might provide for them on inspection.

One difficulty in the interaction is the dislocation caused by the phone rarely being near the device, requiring to-ing and fro-ing to find out more information or even flat refusal to troubleshoot.

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 recorded and made available in a systematic fashion rather they are often only uncovered in an incidental fashion. For example, they might be occasioned by customers comments, such as ‘we’ve had so many problems

---

[1] For legal reasons only the operator side of telephone conversations could be recorded on audio. Customer utterances were recorded in the field notes.

with this machine', 'the engineer was just out last week' and so on. 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 machine' well enough to provide accurate information. For the troubleshooters, information which might be useful to the diagnosis is frequently not available and at best only partial. To add to this customers may report having already attempted to troubleshoot prior to calling the help line and may resist troubleshooters requests to repeat those actions. Even where customers do claim to have performed certain actions troubleshooters will often persist with requests to get the customer to repeat these actions. They may try to avoid interactional difficulty by, for example, differentiating their instructions from the customers previous actions e.g. tagging on additional potentially ameliorative actions, such as cleaning the sensors as well as checking for stuck paper. Troubleshooters do this partly for organisational reasons and partly to ensure the instructions have been carried out correctly. In the absence of other information there is some possibility that the customer may have only partially done something, etc.

The use of diagnostic resources

While working up the problem, its context and history, the troubleshooters can engage with a number of different resources to support diagnosis. The main resource is a troubleshooting knowledge base. In addition to this, troubleshooters have access to online and paper manuals, screen shots and menu maps, models of different devices and their colleagues. Troubleshooters did not always use the knowledge base as intended, that is, to enter the customers problem, drill down from the results to the relevant case (problem description) and then direct the customer through the suggested solutions in order. Troubleshooters sometimes used the knowledge base as a memory aid, going only as far as the list of cases or solutions and using these to prompt questions and instructions to the customer. Alternatively, they used it purely as an accounting device, looking up the solution for its code after diagnosis had finished. To use the knowledge base effectively the troubleshooters need to know how to search it and to identify the relevancy of the results. One part of this work involves translating the customers descriptions into the technical terms of the knowledge base, as well as being able to translate their own search terms into alternatives, where they are not effective.

Troubleshooters have to co-ordinate the use of the knowledge base and their interaction with the customer, despite that text and talk are not always comfortable bedfellows. Often troubleshooters begin searching while the customer is still talking and they often anticipate information they will need for the knowledge base, asking standard questions such as weight of paper and so on. These are relevant questions yet allow time to get to the appropriate point in the knowledge base as they are being asked and answered. When this anticipation is not possible or it is necessary to actually 'read' the text, troubleshooters have to interrupt the interaction, putting the customer on hold. The cases and solutions in the knowledge base 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 copiers, parts and various symptoms and problems that enables them to identify the correct path through the knowledge base with relative ease once they have managed to carry out a successful search. At this point we can see the dual aspect to the translation work the troubleshooters perform, as they translate the instructions into terms the customer can understand, at the same time as situating them within the interaction. There is not one single standard for translation, rather, the troubleshooters adapt their language to the customers exhibited competence.

Troubleshooters may also ask colleagues for advice, although they only usually resort to extending the interactional cohort when they are have run into a dead end - because it is time consuming and requires the customer to be put on hold .

Resolution

Although one might consider diagnosis and resolution as separate activities, i.e. first one finds out what the problem is (diagnosis) and then one fixes it (resolution), in office device troubleshooting, there is not always a clear distinction between diagnosis and resolution. 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. For example, jams may be caused by damp reams of paper, small bits of paper blocking part of the machine, dirty sensors, etc. If the machine works after inserting a new ream of paper or cleaning the sensors, one can then make assumptions to the causes of the problem. However, in the troubleshooting sessions we observed such causes are not routinely provided as part of the diagnosis, rather they are invoked for specific purposes – educating the customer, so they can prevent or solve this problem in the future; demonstrating the professionalism of the troubleshooter, by showing the advice to be good and relevant advice, etc. Troubleshooters are *accountable* to the customers, they are not taken to be implicitly trustworthy in their advice but have to provide the customer with the means to see that their advice in this situation merits such trust. Troubleshooters make use of their expertise to provide justifications for their instructions. On the other hand, the customers to are rendered accountable for troubleshooting by the troubleshooters. The troubleshooters are skilled at persuading unwilling customers to troubleshoot, using minimising terms and requests that are interactionally difficult to refuse (Sacks, 1992) e.g. 'could you just try a little something on the machine for me?'

Just as not having direct access to the machine requires interactional work during the problem description stage, it can cause interactional difficulties during problem resolution. Troubleshooters need to situate their instructions in the ongoing interaction between themselves and the customer, but also between the customer and the machine. However, they only have limited resources for understanding the customers interactions with the machine, what they can hear and what the customer tells them. Often the customer can either be on the phone or interacting with the machine, since frequently phones and machines are not collocated and even where they interacting with the machine requires the use of the customers hands. Whilst description of physical actions often entails some kind of enactment of those actions as they are described, the physical object upon which the actions should be performed is not available to both the troubleshooter and the user. Troubleshooters therefore have a number of techniques and tools for embodying the solution from miming as they describe, to using images and menu maps to going to the machine itself. In all these cases, although they can help the troubleshooter visualise the problem, they require enacting the actions remotely from the device and then describing them and the representations of the device are just that, they do not represent 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. For other discussion of the study see O'Neill et al, 2005a; O'Neill et al, 2005b; Crabtree et al, 2006

TECHNOLOGICAL INSPIRATIONS

The ethnography emphasised the collaborative nature of diagnosis; how the balance of expertise and access to the device is managed by coordination work to mediate between the participants asymmetrical access to the resources necessary for problem resolution. This understanding led to some inspirations for technology design to support such work. The troubleshooting context outlined above, involves a workflow where the end-user can access and perform operations on the device and communicate with a remote troubleshooter who has access to a knowledge base and other resources. This flow is described in Figure 1 with the vertical and horizontal arrows. The observed issues are the dislocation between people and resources but also the difficulty of communications between the end-user and the troubleshooter. In this paper we describe a technology solution which aims at better supporting this interaction. However, there are also other places where interventions might be useful. It is costly (to both customer and service organisation) to have to contact a call centre for support and can be time consuming. Thus we also consider other places where interventions might be useful: online and directly on the machine itself. Whereas in the troubleshooter-customer workflow, difficulties may arise because of the dislocation between the problem and the site of problem resolution and by the remote nature of work, there is at least a human expert, who with all their intricate social knowledge, can dynamically adapt the available resources to the situation as required and so on. When remove the human from the equation we lose a lot of this adaptability. We therefore wanted to consider how to bring some of the critical features of the social into the interaction with machine and online system. Our technological solutions are based around the concept of mediation, that is the technology is always at the centre of the troubleshooting environment to mediate between the user and the various troubleshooting resources (be they the knowledge base, the machine and even the troubleshooter). In this way the system can mediate directly between a person and a remote resource (diagonal arrows in the figure).

Following these considerations, we have designed and partially developed some technology solutions, each of them addressing some specific aspects of the troubleshooting task. They are described more in details in the following sub-sections, beginning with customer access to the online knowledge base, through customer access to troubleshooting on-the-box, to better supporting the customer-troubleshooter interaction.

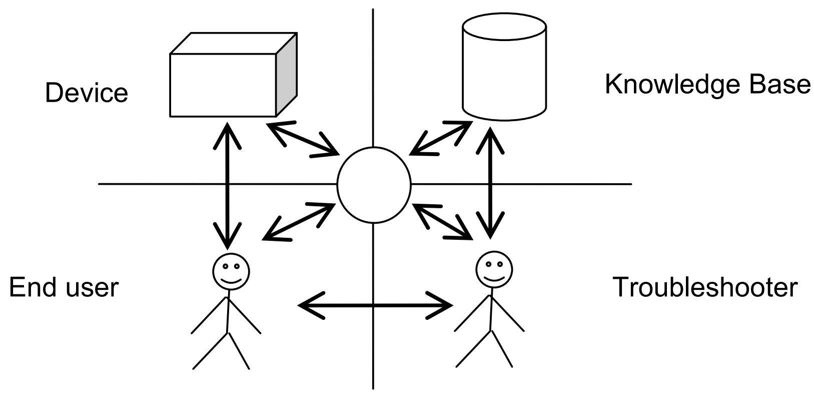

Figure 1 Supporting interactions in the troubleshooting process

ONLINE SUPPORT

Customers have been given access, via the internet, to the same troubleshooting knowledge base that the troubleshooters use, with the idea that they can solve their device problems without needing to call the call centre. The existing troubleshooting knowledge base has a rather basic search mechanism, a poor refinement mechanism based on static predefined categories which did not always fit with the users understandings of their problems and often produced long lists of possible solutions with users rarely looking beyond the first 14 or so. Our aim was to enhance this troubleshooting support, to enable the users to reformulate their queries, based on dynamic refinement choices, to navigate through results sets with greater ease and to enable translations between user and knowledge base terminology. This was based on our understanding that customers rarely come to troubleshooting with fully formulated problem descriptions, in the language of the knowledge base, and that users need help to translate and understand device and knowledge base terminology. It should be noted that improvements in online support may also help the troubleshooters who themselves, despite a greater array of possible technical terms and greater device knowledge, can have problems locating the right solutions.

We have developed an NLP-based interaction layer, called Pocket Engineer, to support the translation of the customers language into that of the knowledge base, using a rich synonymy mechanism, and the navigation of the troubleshooting solutions space, providing a flexible search mechanism based on the extraction of salient features from the knowledge-base contents. The Pocket Engineer (PE), acts as a text retrieval system tailored to searching in troubleshooting knowledge bases. It supports troubleshooting via an interface that provides both guidance, in the form of an iterative query refinement mechanism that prompts users with choices they may not have thought of on their own, and flexibility, in the form of free-text search functionality that lets users who know exactly what they are looking for go directly to the results. The refinement mechanism uses a natural language processing component to generate meaningful choices automatically.

The troubleshooting content that PE accesses already existed before the PE project began, and represented a considerable investment of authors' and editors' time. An important design constraint for PE was that it should function as a new means of access to the existing content, without requiring any additional intervention by subject matter experts or editors. This constraint ruled out approaches such as decision trees which would require subject matter experts to write and maintain additional resources in addition to the textual content. We hope that the result is an access layer that will be easily adapted to other document collections.

The PE interface consists of three sections in which different actions can be performed (see Figure 2).

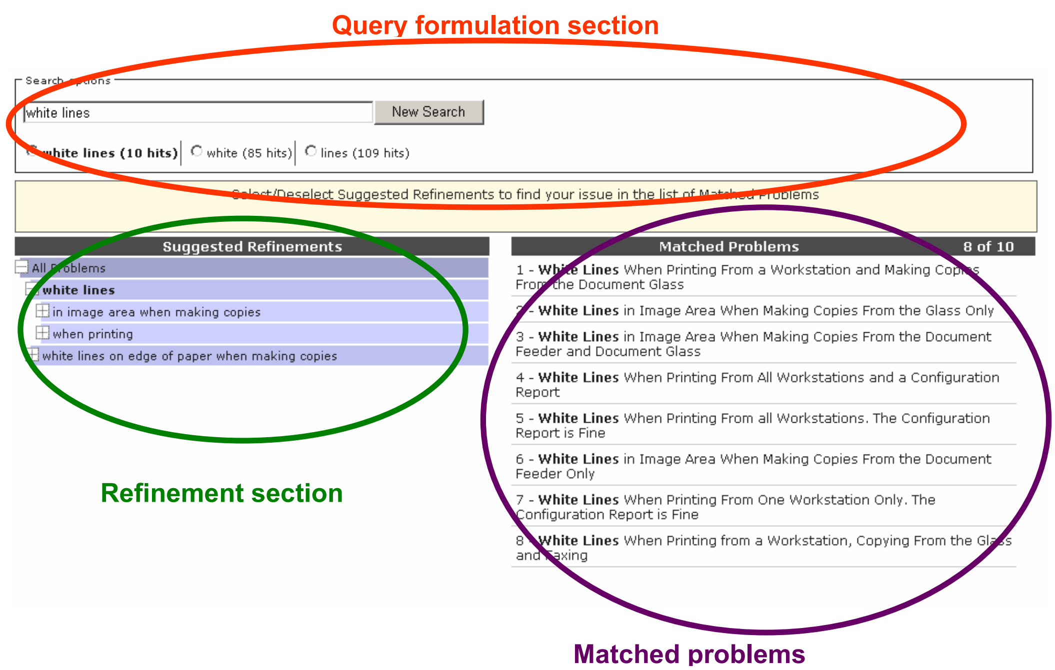

Figure 2: The PE Interface.

At the top of the window is the query formulation section. The user can enter a new query in the query box, or modify a previously-entered query. After the search, the system displays a list of sub-queries for which there are results. The sub-query that the system predicts to be most appropriate is pre-selected, and the results and refinement possibilities for that sub-query are displayed in the lower part of the interface. This reduces the results set that the user is directly exposed to and shows the user exactly which of their query words have been matched by the system. The user can click on other sub-queries to see the other results.

On the lower left-hand side of the interface is the refinement section. This consists of a dynamically constructed tree, whose nodes are expressions extracted from the knowledge base. On the lower right-hand side is a list of problem descriptions, which changes as the user clicks different nodes of the refinement tree. The root of the refinement tree is labelled "All problems"; clicking on this node causes all problem descriptions that match the sub-query to be displayed. The children of the root are expressions from the knowledge base in which query words (or synonyms thereof) appear. After this first level of the tree, each node is an expression that adds further detail to its parent expression. For example, 'white lines' has the children 'in image area when making copies' and 'when printing'. Clicking one of the nodes of the tree has two effects: it causes the list on the right to be reduced to only cases that match the refined problem description, and it opens a new layer of the refinement tree in which further refinements that differentiate among the remaining cases are proposed (see Figure 2). The refinement mechanism helps the user explore the problem space, by showing which query terms have been matched and by helping users make their query more precise. Since the refinements are dynamically constructed they reflect the query that the user entered.

A rich synonymy mechanism helps mediate between the user's terminology and the terminology found in the knowledge base. It allows synonyms to be defined across phrases and with contextual rules, e.g. document might be a synonym of 'page' unless adjacent to 'feeder', when it is a machine part 'document feeder'. This will enable users to search using as much or as little of the technical terminology as they have available. Non-textual representations can be associated with terms that might be unfamiliar to the user, helping the user to understand and learn the terminology of the knowledge base.

Details on how the system dynamically generates a refinement tree for a given query and the rich synonymy functionality that PE provides to improve both search results and the refinement tree can be found in [ECIR07]. However, providing support on line does not solve the current dislocation between the machine and the customer, as PC's like phone are rarely by shared machines, the next solution attempts to address this.

END USER-MACHINE

In this solution access to the knowledge base is provided on the machine. Devices already provide some troubleshooting support. Typically suggesting operations to be performed or providing warnings, when a problem that can be detected by the device arises. Although useful, this kind of help is often quite limited, to only a few situations and it does not cover user 'how to's'. We therefore aim to enrich the existing interaction available on the devices with the layer for direct access to the knowledge base. It will both solve the observed issue of dislocation between the device and the user when accessing remote resources and reduce the distance between the perception of the problem that the user has and the characterisation of the problem that can be done by the device according to what it is able to detect.

We are currently working on a design which takes into account the dimensions of the device display and the expectations of the users when interacting with the machine, as well as leveraging the potential device information about the problem, the current state of the machine and so on. Such information could be used to contextualise the troubleshooting searches. As well as addressing the dislocation between site of problem and support resources this solution could be used to make the customers accountable for troubleshooting their machine. Just as the troubleshooters persuade the customers to troubleshoot, the device could pull the customer into troubleshooting by suggesting solutions they can try when problems occur.

Another dimension of the device-end user interactions that can be exploited is the study of these interactions over the time. Office devices, e.g. printers, are often shared resources utilized by multiple users, who may attempt to solve device problems, using experience, instructions from the machine or the knowledge base. Currently, to our knowledge, printers do not leverage knowledge about past interactions with the machine. For example, when a problem occurs and a number of users follow the instructions provided by the machine but are not able to solve the problem, successive other users may each try to solve the problem without knowing of the prior failed attempts. We have developed a method which leverages this knowledge to help people avoid trying out the same unsuccessful sequence of troubleshooting actions on a broken device. The method allows a user who encounters a problem at a printer to be informed if other users have already tried a sequence of actions.

Our proposal involves the recording of all user interactions with the machine in particular those linked to troubleshooting and maintenance operations. These interaction data include: 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, for e.g. a troubleshooting knowledge base, to understand: (1) if some users have somehow (e.g. only partially) followed the troubleshooting instructions given by the machine 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* [CanceddaEtAl.03, LodhiEtAl01] to match ideal sequences with real sequences taking account of the noise. The information can then be provided to the user, X users have tried Y action, or can be used to inform which solutions are shown to the user, e.g. if doing Y doesn't solve the problem, try Z.

COLLABORATIVE TROUBLESHOOTING

The final design proposal aims to address the problems that the customers and troubleshooters have when collaboratively troubleshooting over the phone, that of dislocation, embodying the solution, knowing what the customer is doing and asking the customer to repeat prior actions. This solution provides the troubleshooter with direct access to the device status and improved communication through the use of visual artefacts. A bi-directional shared representation (BDV) of the troubleshooting problem is provided, available to troubleshooters, to 'see' users actions on the machines, and to the customers, so that the troubleshooters instructions can be visualised. Such a representation would provide a resource for coming to an understanding the problem and mutual orientation and interaction. We choose a representation rather than video because it could provide a ready recognisability of what is shared or purely local and low equipment overheads.

The problem representation will consist of a linked 3-D model of the device and a number of means of interacting with this model. The BDV will be presented on the device at the local site (on the kinds of medium sized screens increasingly available with modern devices) and on the troubleshooters terminal at the remote site. It is *linked* to the device, such that actions on the device are shown on the representation, e.g. if a user opens a door, that door will appear open on the representation. This is enabled through the many sensors that already reside on such devices. In addition both the customer and operator are able to indicate parts on the machine, and the operator is able to demonstrate visually actions which should be performed (for example, lifting a handle and sliding a toner cartridge out of the machine). The customer will access technical support through audio-visual communication channels located on the machine itself. The audio channel will enable the customer to converse with the operator. The visual channel will show the BDV. Thus the machine becomes the infrastructural mediator between users and technical support. The BDV enables *both parties* to have a real time understanding of the actions which are being or should be performed on the machine. These provide a resource for overcoming the troubleshooting problems we have described.

To illustrate this design proposal we describe a typical scenario showing what the user and troubleshooter would see and could do:

* A user discovers a problem with their device they cannot fix themselves. They place a call to the support centre by selecting a 'call support' button on the device UI. The device

establishes a secure data and audio end-to-end connection to the support centre. Data might include the device serial number, sensor information on current machine state, historical information, etc

* A remote server uses the data in combination with its own stored data for that device - e.g. records of other troubleshooting sessions - to build an initial representation. This representation is displayed on the operator display and on the customer interface.
* The customer-oriented version of the representation consists of a visual representation of the current status of the device, showing which doors are open, which trays are empty and so on.
* The initial representation for technical support will contain information on the current status of the device including a 3-D representation of the device that corresponds with the customer's and the means to view this representation from different spatial perspectives. Such perspectives will facilitate at-a-glance recognition of problems. In addition the operator will have control buttons on the representation through which they can interact with the customer, e.g. to *indicate a part* of the machine, *select an action* the user should perform and so on. Then, the action is shown to the user on the display of the device and, if visual indicators are available on the device, the visual indicators of the selected part are activated.

The customer meanwhile can indicate parts of the printer, perform actions *on the machine* which are shown on the shared representation, and view the operator's interaction with the representation when not on hold.

The BDV would help troubleshooters to situate their instructions in the stream of activity as the operator 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 so that the haecceities, the 'just thisness' of the problem (Garfinkel and Weider, 1992) can be explored and revealed.

Acknowledgements

We thank A. Kaplan, C. Roux, K. Pettersson, and J. Selin for their contributions to the Pocket Engineer and the design of the on-the-box interaction, and N. Cancedda for his contribution on the method to avoid unuseful repeated troubleshooting interactions on devices.

References

[CanceddaEtAl.03], N. Cancedda, E. Gaussier, C. Goutte and J.-M. Renders, "Word-Sequence Kernels", Journal of Machine Learning Research, 3(Feb):1059—1082, 2003.

Crabtree, A., O'Neill, J., Tolmie, P., Castellani, S., Colombino, T., & Grasso, A. (2006) The practical indispensability of articulation work to immediate and remote help-giving. *Proc. CSCW*. 219-228

[ECIR07], F. Roulland, A. Kaplan, S. Castellani, A. Grasso, C. Roux, J. O'Neill, K. Pettersson, "". In Proceedings of ECIR, 2007, Rome Italy.

O'Neill, J. S. Castellani, A. Grasso, P. Tolmie, and F. Roulland, (2005a) "Representations can be good enough", In Proceedings of ECSCW.

O'Neill, J. S. Castellani, A. Grasso, P. Tolmie, (2005b) In Proceedings of Interact, Rome, Italy.

[LodhiEtAl01], H. Lodhi, N. Cristianini, J. Shawe-Taylor and C. Watkins, "Text Classification Using the String Kernel", in Advances in Neural Information Processing Systems 13, MIT Press, 2001.

Sacks, H. (1992) Lectures on Conversation, Volumes I and II. G.. Jefferson (ed). Blackwell, Oxford