

Using Real-Life Troubleshooting Interactions to Inform Self-assistance Design

Jacki O'Neill, Antonietta Grasso, Stefania Castellani, and Peter Tolmie

XRCE, 6 Che de Maupertuis, 38240 Meylan, Grenoble, France
{name.surname}@xrce.xerox.com

Abstract. Technical troubleshooting is a domain that has changed enormously in recent years. Instead of relying on visits from service personnel end users facing technical problems with machinery, for example computers and printers, can now seek assistance from systems that guide them toward an autonomous solution of the problem. Systems that can be offered to them are wide in their range, but typically fall either in the category of Expert Systems or searchable databases that can be queried with keyword searches. Both approaches present advantages and disadvantages in terms of flexibility to address different levels of user expertise and ease of maintenance. However, few studies explicitly address the issue of how best to design for a balance between guidance and user freedom in such systems. In the work reported here an office equipment manufacturer's call centre was studied in order to understand the mechanisms used when *human* agents guide users toward a resolution. The overall aim here is not to reproduce the agent behaviour in a system, but rather to identify which interactional building blocks such a system should have. These are assessed in relation to the existing online knowledge base resources offered by the same company in order to exemplify the kinds of issues designers need to attend to in this domain.

1 Introduction

In recent years a number of systems which support end-users in self-troubleshooting problems with their machines have become available. This is especially noticeable amongst companies where there is a heavy service commitment and a concomitant desire to minimize the costs associated with that [5] [8] [9]. There is therefore a strong impetus towards providing better tools for users who are willing to try to solve autonomously the problems they face. In large part troubleshooting activities are based on using technical expertise to identify causes and ways to proceed beyond the evident symptoms people might be witnessing. Because of this many of the proposed tools and systems are based on Artificial Intelligence techniques, which allow the machine to embody a model of the domain and let the user navigate within it, even where they lack any deep technical knowledge. Examples of such systems are the SACSO system [6], based on Bayesian networks and the NaCoDAE system [1], based on Conversational Case Based Reasoning. A wide review of Intelligent Help Desk systems is provided in [7].

These systems, while different in the particular models they choose to use, nonetheless all share the assumption that to build a model based upon their explorations of the domain is the correct way to proceed. Thus they all exhibit an engine that embodies a description of how symptoms and faults could be correlated that is directly derived from such a model. This underlying modelling presents several advantages, for example offering the possibility to guide the users toward the solution through a number of fixed steps. However, these AI-based systems often appear too rigid, since the interaction model often forces the user to follow the predefined steps without easy shortcuts. Sometimes motivations such as wanting to make one system fit all and keep down the costs of different implementations results in the same system needing to address a range of expertise from qualified experts to novice users. In such cases the rigidity of an AI-based system can be even more problematic. A useful example of this kind of problem, where a case-based reasoning system proved to be inadequate for the use of call centre troubleshooters is presented in [10]. At the opposite end of the scale, following advances in full-text Web search engines, systems have been proposed that are based on Information Retrieval techniques where the content is structured in terms of cases and solutions. An example of such a category of systems is Eureka [3] where a database of tips for service engineers can be accessed through the entry of a query. Benefits of the latter approach appear to be the possibility for a user to express a query in his or her own words, together with the possibility of freely navigating the content without a rigid structure. An additional benefit of such systems is that they can easily be dynamically modified by field engineers or troubleshooters, in order to insert new content expressed with their own wording and fitting exceptional cases that were not originally taken into account by the knowledge base designers. Aside from a number of drawbacks to such solutions we will be discussing throughout the course of this paper, other possible downsides are a certain redundancy and heterogeneity of style in their contents.

In addition to the above some systems have been proposed that make use of both the AI and the Information Retrieval approaches. This is the case with the MCRDR system discussed in [7], where a case-based reasoning system is complemented with keyword search to provide both quick navigation and guided interaction when the user is unclear about appropriate keywords. This latter approach is certainly promising as it seems to offer the best of the two approaches: guidance and flexibility.

In this paper we aim to contribute to the design of such hybrid systems by drawing upon real-life observations of troubleshooting activities and the insights that have grown out of them. In particular we will be proposing that design in this domain needs to properly attend to the way such interactions currently proceed. This is not to be undertaken simply to replicate every feature of current human-human troubleshooting interactions. Rather, it is that this is the only way to effectively identify the interactional building blocks that a system *will* have to support if it is going to truly meet the requirements of online users.

In order to meet this objective we investigated both the keyword based online Knowledge Base of a leading supplier of office equipment and the telephone interactions between troubleshooters and users at its European support centre. Our initial step here was to gain access to customers' troubleshooting practices and understandings by undertaking an extended ethnographic study of the work of troubleshooters (see [3] and [4] for an exposition of the rationales for conducting these kinds of studies in a design context). The troubleshooters in question work through problems and the solutions with

customers whilst at the same time drawing upon a knowledge base to underpin their expertise. The knowledge base we are referring to here has recently been made available to customers so that they can troubleshoot for themselves without calling technical support or an engineer. It is typical of many databases, consisting of a repository of information which the customer accesses through a word search. Although troubleshooters have access to some additional materials the knowledge base used by both troubleshooters and online users is to all intents and purposes the same and utilises the same search mechanisms. Analysis of the work of the company's troubleshooters therefore offered two important elements. First of all it gave us access to the methods and reasoning customers and troubleshooters engage in together to successfully troubleshoot failed devices. Secondly, it gave us the opportunity to explore the ways in which troubleshooters might be providing a resource for successfully *working the knowledge base* that stands above and beyond the knowledge base itself. Our other activity was to explore how *non-expert* users engage with the company's knowledge base in order to try and fix problems for themselves. This offered us a way to both compare on-the-phone and self-assisted troubleshooting with access to the same online resource, and to assess the ways in which the knowledge base currently provides support in its own right.

In this paper we will be focusing on the ways in which observations derived from these studies have highlighted fundamental interactional features of troubleshooting that have clear implications for how the existing keyword-based system could be enhanced to support its online users. To do this we will begin with a brief outline of the online resource. This will be followed up with a discussion of the various findings from our study, findings that indicate a number of issues to do with the tension between people's practices and the existing organisation of the knowledge base. These will be shown to centre upon the character of problem descriptions, terminology, the ways in which problems and causes get constituted, and how solutions actually get located and implemented. This will be followed by a number of suggestions for how the design of systems might be informed by these insights and might locate ways of supporting both structure and flexibility at the same time.

2 The Knowledge Base

The troubleshooting knowledge base we have been referring to here is an online resource provided to the customers of a major office equipment supplier in order to facilitate the resolution of problems with their machines without recourse to phoning a troubleshooter or calling out a service engineer. It is made accessible from the support pages of the company's website, and both customers and troubleshooters can use it as a resource for finding solutions to their technical problems.

The knowledge base is organized around products, each providing a similar entry page where two main troubleshooting areas are provided: search area and a sidebar with links to related resources (see Figure 1). Using the search area customers can issue searches using multiple keywords. Tips for searching are also provided on the page, giving advice about how best to formulate queries, e.g. 'Install toner cartridge', 'Paper tray not recognized', and 'Read the meter'. In addition to this main interaction area, a sidebar provides links to resources like procedures and manuals, as well as an entry point to the content of the knowledge base for pre-categorized problems.

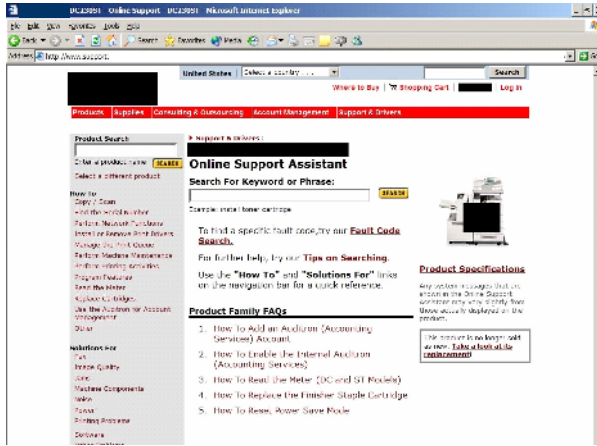


Fig. 1. The Troubleshooting Knowledge Base

3 Users' Practices and the Organization of the Knowledge Base

As we have already outlined, the customers of the office equipment company are provided with both an online knowledge base and access to one of several large support centres which they can telephone in order to request help from troubleshooters. The troubleshooters they are passed on to then seek to establish the nature of the problem and talk the customer through to possible solutions. The cornerstone of our initial investigations into the possible issues in this domain was a three week ethnographic study of troubleshooting work in one of these support centres, responsible in this case for the whole of Europe. Troubleshooters across a range of teams were observed intensively throughout their working day in order to be able to record and acquire a deep understanding of how troubleshooting is accomplished in actual working practice. Data collected included field notes, audio records of the troubleshooters side of the conversation and video of the troubleshooters interaction with the knowledge base and other tools. Multiple interactions with customers, across a range of problems and products, were witnessed from beginning to end, and analysed in order to uncover the way those troubleshooting interactions are organised. Out of this we evolved a clear picture of both how troubleshooting stands as a methodic accomplishment, and the problematics that the actual practice of troubleshooting over the telephone must overcome. One example of such a problematic is that customers are often not by the machine when they call in and in many cases the only solution is to give piecemeal instruction as a customer goes to and fro to their machine. Another problematic of this order is the need for troubleshooters to resolve their own inability to see the troubled device. In this paper, however, we shall be concentrating upon the implications of how troubleshooting is organised as an interactional practice for those situations where people can only interact with *troubleshooting systems* rather than other human beings. In the course of this we shall be directly considering how these might relate to the same company's existing provision of such support, their online knowledge base. These considerations are underpinned by

our own observations of the use of that knowledge base by non-expert users and of its use by expert troubleshooters in the support centre.

3.1 Symptoms, Causes, Faults and Solutions

Our first observation here relates to the way in which customers encounter and describe problems with their devices in terms of symptoms. When they make contact with troubleshooters they then work through the problem together making use of symptoms *and* causes. Symptomatic description is a commonsense way of reasoning about a trouble in the absence of the necessary technical knowledge, e.g. ‘the copier machine isn’t stapling’; ‘the machine blocks up with paper’. Symptomatic understanding can be contrasted with the technically oriented ‘fault’ understanding displayed in the knowledge base. The following is a simple example offered to illustrate the distinction between faults and symptoms. When taking a car to a garage there are some faults an owner can just plainly see and report to the mechanic. For example, it is possible to see that the headlight glass and bulb are smashed. However, owners often also report things like ‘it makes a knocking noise when I go round the corner’. The owner may have no idea what fault could cause such a noise so they report the symptom. Not possessing a technical understanding of cars and their faults, the owner resorts to a commonsense symptomatic description of the car’s adverse behaviour. With sufficient knowledge they might well describe the fault in the same way as for the smashed headlight. So, the very fact they describe the matter symptomatically displays the absence of such knowledge, making it unlikely they would be able to reason through from the symptom to the cause. The mechanic, of course, when asked by a colleague about the same problem, will not in all likelihood answer ‘it makes a knocking noise when it goes round a corner’. Instead he or she might say something like ‘the kingpin is worn’. The very act of describing the fault here displays a technical knowledge of cars.

When using the knowledge base customers are being asked to act like the car mechanic and make those orders of interpretations without necessarily possessing the relevant expertise. In one exchange during our observations we witnessed the following description:

Customer: it screws it up ... in the paper tray.

However, when using the same starting point and searching for ‘screwed up paper’ in the knowledge base, it brings up 42 results, the first seven of which are shown in the screenshot in Figure 2.

None of these obviously describe the problem, but it is possible that 1, 2, 3, or 4 might be the issue. However, from neither the titles nor the solutions beneath them is there any information on what the printer would do when it has that problem. That is, would it screw up the paper? The only way to discover whether one of these is applicable is to try all the suggested steps for each of these to see if they solve the problem.

Troubleshooters work to *mediate* between the symptomatic reasoning deployed by customers and the organisation of the knowledge base. With customers they will happily engage with and use symptomatic accounts themselves, but when it comes to using the knowledge base they use the requisite technical search term that they believe may lead to the fault giving rise to the symptom. For example, when a customer described a squealing noise from her device the troubleshooter was able, through further symptom

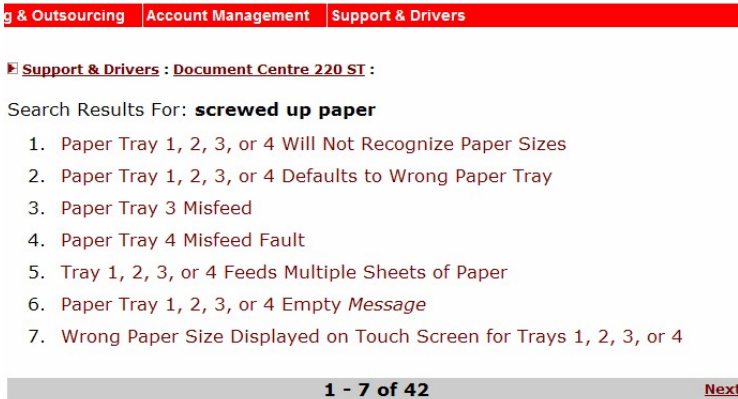


Fig. 2. First seven results for “screwed up paper”

elicitation, to come to the conclusion that the fault resided in the processor. By entering ‘processor’ into the search field the troubleshooter was able to bring up the requisite solution. Utilising their technical knowledge of devices troubleshooters are able to turn everyday commonsense symptomatic descriptions into the necessary technical vocabulary of the knowledge base. Troubleshooters are then able to select the most appropriate terms to enter into the knowledge base from the multiple symptoms described by customers. At the same time, troubleshooters are able to translate knowledge base instructions into symptoms which the customer can look for. They mediate between the commonsense, symptomatic *and vernacular* understanding of problems displayed by customers and the technical understanding of machine faults displayed by the knowledge base.

3.2 Vernacular and Technical Terminology

As we described above customers use vernacular terminology whereas the terminology of the knowledge base is highly technical. Machine parts are referred to by their technical names, e.g. ‘duplex module’, ‘finisher’, ‘upper paper path’, ‘document feeder rollers’, and so on. In addition, technical terminology is also used to describe the faults that occur, such as ‘spooling’, ‘postscript errors’, or ‘guest authentication pass code lockout error’. When customers phone in the troubleshooters can act as translators between them and the knowledge base. However, if customers are using the knowledge base on their own they must already have the right kind of terminology available to them. This applies to both searching the knowledge base and interpreting the results of their searches. The use of a technical vocabulary is not limited to obviously technical language; it is also manifest in the fact that vernacular terms are given a technical meaning in the context of the knowledge base. For example, what is the difference between a black line and a black band? When does a line become a band, and vice versa? The importance of this question can be seen if one uses ‘lines’ to refine a search within the ‘image quality’ field offered by the knowledge base. This brings up 45 results, towards the end of which one finds results that include the term ‘bands’. However, searching for ‘bands’ instead brings up five results all of which relate directly to bands. Lines and bands are obviously

considered different image quality problems as can be seen when selecting ‘Black Lines on Copies From the Glass Only’ which produces the following solutions:

The screenshot shows a navigation bar with three tabs: 'ng & Outsourcing', 'Account Management', and 'Support & Drivers'. Below the navigation bar, there is a search result for 'Support & Drivers : Document Centre 230 ST :'. The search results are for 'Black Lines on Copies From the Glass Only'. The solutions are ranked by relevance and listed as follows:

1. Use a Good Original
2. Clean All Document Glass Surfaces
3. Set Auto Centre Shift
4. Set Border Erase
5. Replace the Copy / Print Cartridge and Make 30 Copies of a Blank Original

At the bottom of the search results, there is a link: [Continue Browsing "Image Quality"](#)

Fig. 3. Results for Black lines

Selecting ‘Black Bands on Copies From the Glass Only’, however, leads to just these three solutions instead:

The screenshot shows a navigation bar with three tabs: 'ng & Outsourcing', 'Account Management', and 'Support & Drivers'. Below the navigation bar, there is a search result for 'Support & Drivers : Document Centre 230 ST :'. The search results are for 'Black Bands on Copies From the Glass Only'. The solutions are ranked by relevance and listed as follows:

1. Clean All Document Glass Surfaces
2. Change the Image Quality Setting
3. Replace the Copy / Print Cartridge and Make 30 Copies of a Blank Original

At the bottom of the search results, there is a link: [Return to Search Results](#)

Fig. 4. Results for black bands

Only some of the solutions are the same for both of the options, thus lines and bands have a precise and different meaning in the technical parlance of the knowledge base. Reasoning technically about the matter may enable an engineer to distinguish different faults and the different appropriate actions to be taken depending upon whether the mark is a line or a band, but that distinction is not one that is available to customers in their everyday encounters with technology. Here, what turns on the distinction between a line and a band may not be readily apprehended.

3.3 Constituting Problems and Causes

The customers and the troubleshooters have to build up a description of the problem and its range of causes between them because the problem the customer is encountering is not immediately available to troubleshooters. Instead they have to elicit the character of the problem through their interactions with the customer and the customer’s reported

interactions with the machine. Customers do not simply describe problems. Instead a customer's telling of problems may have several aspects. They offer up symptoms that need to be resolved with technical descriptions. Wrapped around such symptoms there will be other information that may be redundant from a troubleshooter's point of view. In addition, problems may have multiple symptoms, which the customer may not necessarily tell all at once. For example, in the following extract C describes that there's a jam, that the paper is not going in properly, that it doesn't sound right, and that it is making a noise.

Troubleshooter: Hi Shan how can I help you then
Customer: it's the high capacity feeder
TS: Yeah
C: it says there's jams but there's not
TS: Right
C: When we put the paper in
TS: Um hmm
C: and it's as if it is not going in properly
TS: Yeah
TS: You You're talking about where the tray actually lifts itself up doesn't it?
TS: Um and can you tell me when you look in is the tray still lying flat or is it er a bit off does it look as though it's skewed by any chance?
TS: It does look? Ok I just wonder if we can sort that out now
C: And it doesn't sound right
TS: Yeah
C: It's making a noise

In order to constitute problems and potential causes the troubleshooter has to work with the customer to both reconfigure symptoms in appropriate ways and to uncover from within the information being reported the features that would seem to be most relevant. Troubleshooters orient to these commonsense symptomatic descriptions of problems frequently asking questions to elicit additional, or more precise symptoms such as "what's the noise like", "what's the paper coming out like", etc. That is, they will readily use symptomatic descriptions to refine the nature of a problem. In addition the troubleshooters work to refine their understanding of the customer's problems by asking 'when' or 'where' questions which may require the customer to carry out tests on the machine. For example, a commonly asked question is one designed to determine when the problem occurs. For example, if a problem occurs when printing from the glass the operator may ask, 'does the problem occur when using the document feeder as well?'. Customers may need to carry out tests on the machine to answer these questions, having reported only their initial impression of the problem (e.g. they were printing from the glass, so they knew this was where the problem occurred). The answers to such questions help troubleshooters to narrow down the problem space.

If one takes in comparison the resources currently offered by the company's knowledge base they are evidently more limited. Rather the customer locates solutions through searching the knowledge base. These searches can be effected through the use of keywords or phrases or via a side bar which specifies pre-categorised problems such as 'image quality'. This side bar can be used for category searches but these tend to have the disadvantage of producing large numbers of results. As an example, for one machine, selecting 'image quality' produces 113 results. When it comes to using keywords or phrases we have already seen how the technical terminology can have an impact. Furthermore, the knowledge base does not allow for a ready concatenation of

symptoms. Instead, it searches on each symptom individually. If one is searching in a category such as ‘image quality’, adding a new symptom from another category (e.g. ‘noise’) leads to results which are difficult to interpret in the light of the symptoms. Indeed, as we shall see below, the search results often bear little apparent relationship to the search term. Nor does the knowledge base help with the reformulation of problems. It does indicate some questions which could be used to refine the list of causes. However, these are not clearly presented as questions, but rather as options within a mass of other possibilities which may not even all appear on the same page.

3.4 Locating and Implementing a Solution

Customers are not, of course, unthinking automatons blindly following the advice of troubleshooters. Instead they use their own knowledge and understandings of the machine and its troubles to make sense of the troubleshooting procedure. Thus troubleshooters must offer ‘visibly’ good advice to customers that is relevant to the problem at hand. One way of doing this is by talking about causes. Causes may be brought into troubleshooting sessions for a number of reasons, e.g. educating the customer, but also crucially as a means of demonstrating that the advice they are giving is good advice and relevant to the problem. They can be used as a way of persuading sometimes reluctant customers to troubleshoot. In the observations of troubleshooting work troubleshooters were often explicitly asked to account for their proposals in these terms, e.g.:

Customer: why should that work then? Will it clean it?

Troubleshooter: Exactly yeah yeah could be a build up of dust or something like that you know

In another example we can see how a customer is given pause by advice that at first sight seems to contradict their commonsense expectations about what a reasonable procedure should consist in:

Troubleshooter: ‘Ok have you cleaned the pick pad and the pick roller on the machine? ...in the tray there’s a pick pad and pick roller that need to be cleaned basically that’s what um takes the paper out of the tray and if there’s dirt on them then it won’t actually pick the paper out of the tray’

[troubleshooter directs the customer to the location of the pick pad]

TS: Ok? So what we do need to do is um basically clean this pick pad and there’s also a roller then that er kinda comes down on top of it (.) ok?

[troubleshooter instructs the customer in cleaning the pick pad]

TS: yeah Ok and then once you’ve got that done we need to do the very same with the actual roller that corresponds to the pad that takes the pa the paper out ok? So

Customer: Ok let me get a bit of tape

TS: Sure

(pause)

C: OK done that

TS: Ok so what I want you to do now is if you you can put the paper tray back in

C: in?

TS: Yeah you can put the tray back in

(pause)

C: OK

Here the troubleshooter has told the customer that they need to clean both the pick pad and the pick roller which comes down on top of it. Having completed the cleaning of the pick pad the troubleshooter has instructed the customer to put the tray back in. However, the customer queries this instruction, so the troubleshooter has to confirm it.

After a telling pause the customer acknowledges the instruction and proceeds to carry it out. But why query the instruction and hesitate about it? Here the customer can be seen to be actively reasoning, making sense of the instructions on the basis of his/her commonsense understanding of them. They have cleaned the pick pad and are now about to clean the adjacent roller. This, to them, would appear to be a constituent part of the tray assembly, so putting the tray back does not seem to make sense. However, it does make sense once one is in possession of more technical knowledge for it turns out that the roller is accessed via another area of the machine.

Whilst the solutions offered in the knowledge base might be understood to be premised upon the causes of certain kinds of problems these causes are never made explicit. Thus customers are never able to directly relate the actions proposed to what might be underlying the symptoms they have witnessed and are denied another resource integral to the reasoning about problems they engage in. In addition there is the afore-mentioned and more serious problem of the lack of an apparent relationship between the search query and the presented results. Search results, in this case, frequently do not have the keyword(s) used, or even anything obviously related to the keywords. Indeed they often have no visible relation to the problem to which a solution is being sought. For example, if we look back to Figure 1, which was about a search for “screwed up paper”, we can see how it brought up several apparently unrelated results, such as number 6: “Paper Tray 1, 2, 3, or 4 Empty Message”.

However, results which are not apparently related, from the customer’s point of view, may be related from the technical point of view. In this particular case the proposed solution related to the possibility of a type of jam where some fragments of paper affect the sensor detecting the size of the sheets, resulting in a message for an empty tray. Thus, to understand such relationships between search query and search results the customer must have a considerable technical knowledge. If the customer does not possess such knowledge the results appear nonsensical. Even where results obviously relate to the query, more often than not a search will result in multiple options and it is necessary to choose between them. However, it may be that the differences between the options are not, from a commonsense point of view, readily understandable and again require a technical orientation for someone to understand. To illustrate, in Fig. 5 are the results of a search under ‘noise’ for one machine:

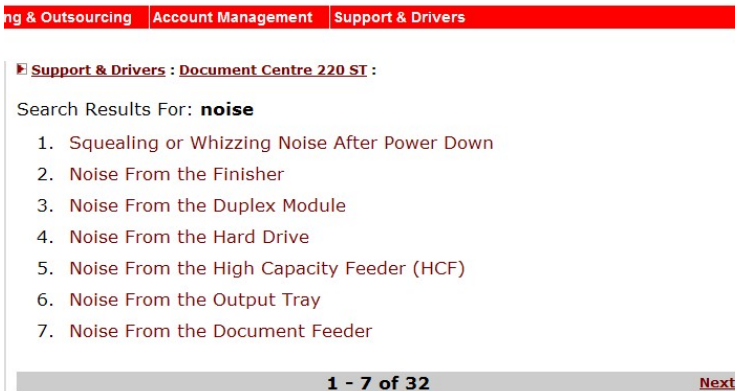


Fig. 5. First seven results for “noise”

The problem with this set of results is that it requires a certain (relatively advanced) knowledge of printers to decide which option to choose. It is necessary to know both what the various parts referred to on the printer are and exactly where the noise is coming from. To address the first issue, the customer could click on the various options in the hope of seeing pictures of them (and thus identifying what, for example, a Duplex Module is). But this is onerous, putting extra work on customers when they could just ring for help. In addition, only one solution here gives any kind of indication of the sort of noise the different problems could make, which could substantially help to narrow down the search for the customer. Indeed in the ‘noise from the processor’ example above the troubleshooter was swiftly able to identify the location of the noise from the customers’ description of its sound. For the customer to select the correct, or even the most likely option from those presented, requires them to be knowledgeable about both the printer and the problem, rather than enabling the customer to explore the problem from the most basic understanding of it. One can see in that case how the organisation of results in this manner does not enable the customers to make use of their common-sense understandings of the problem or even to learn how to couch their problems from a technical viewpoint.

In the above discussion of the work practices of troubleshooters and customers when troubleshooting, and the extent to which these are reflected in the current organisation of this particular company’s online support knowledge base, we have indicated a number of potentially significant concerns regarding the design of knowledge bases for customer-only troubleshooting. In the next section we will outline some design recommendations to address these issues. It is important to note that these recommendations could be applied to the *existing* knowledge base through the creation of a new interface, thereby enabling customers to be better supported in customer-only troubleshooting at a relatively low overhead. As is the case here, many such knowledge bases are already up-and-working and available to customers, making the overhead involved in making changes one of the important ‘real-world’ considerations that must be attended to.

4 Design Recommendations for Knowledge Base Redesign

When we had completed the investigations reported above, we came to the conclusion that the design of keyword based search systems should not simply follow the design of typical Web engines. This is because the vast majority of customers troubleshooting are not able to assess the technical character of their problem or what a solution could look like. They are therefore hugely dependant upon additional guidance and support for the exploration of their problem beyond their constrained understanding of it. We believe that the interaction mechanisms that have been described in this paper can guide the design of such systems by bringing the expertise of the troubleshooters to the user-machine interactions. Using an understanding of these interactions can inform the design of interfaces for knowledge bases, in order to support both guidance and flexibility. This interface would provide mechanisms to better enable the customer to interrogate and interpret the knowledge base and thus solve their problems. This involves taking into account a number of interaction principles:

- * Organisation around a symptomatic taxonomy, where the structure reflects the symptoms and causes of problems, along with their solutions, rather than being

organised on the basis of faults and solutions. Currently troubleshooters translate customers' reported symptoms into search terms appropriate for the knowledge base. As it stands online users have to manage without any such provision. Organising the knowledge base in terms of symptoms and causes would enable a search to be done according to initial symptomatic understandings of the problem.

- * Enabling customers to work up an appropriate problem description within the problem space for themselves out of their own understanding of the problem. Non-textual representations of symptoms could be presented where relevant, enabling easier identification of symptoms than through subjective textual descriptors alone. In relation to this it should be possible to add more symptoms and thereby systematically refine their original problem description. In a similar vein, it should be clarified where 'when/where' tests need to be carried out by clearly showing these options to the customer, thus enabling them to narrow down the problem space through testing the device.
- * Providing the possibility of searching on either vernacular or technical terms or a mixture of both. This will enable customers to utilise the technical knowledge that they have, but where they do not have such knowledge to be able to describe the problems in the terms they *do* have available to them.
- * Providing support for understanding the results of searches, with technical terms being given a lay description or an indication associated symptoms.
- * Presenting search results in a way which clearly displays the relationship between them and original search terms that were entered. For instance, if one of the symptoms of a problem is a noise and this underlies why results are displayed, even if the result titles themselves do not appear to be noise-related, then a list of the symptoms you get with that kind of problem could be provided underneath.
- * Beyond the search terms themselves, making clear how the results relate to the problem the customer is experiencing. One possibility here is that causal information could be provided that would enable a customer to see just why a set of instructions might apply to their problem. Explanations of what carrying out such instructions does would also help. Symptomatic and causal information could assist in establishing a clear link between symptoms/problems and solutions.

5 Discussion

It is evident that there are a number of usability issues with the kinds of purely keyword based knowledge bases for troubleshooting problems we have been discussing in this paper. It is also evident that on-the-phone troubleshooters do a lot of work to mediate between customer locutions and the content of such resources. This latter observation carries particular force when one considers the fact that customers have to use the knowledge base on their own, deprived of the very resources and competences that provide for the knowledge base being an effective working tool. In that case there is a clear need for self-assisted troubleshooting systems that attend to and articulate around the critical interactional concerns outlined above. To review the issues here: 1) The information in the knowledge base is organised in terms of faults and solutions rather than symptoms. This makes it necessary for customers to have a high level of knowledge about their devices. 2) Troubleshooters currently translate customer symptoms into

search terms and, where necessary, enrich this with their own understanding of causes. This is something customers are not necessarily well-equipped to do on their own. 3) On-the-phone customers often report more than one symptom and troubleshooters work to take these into account. However, the knowledge base itself does not provide an intuitive way of collating symptoms or incorporating new symptoms into a search. 4) Troubleshooters work with customers to elaborate and refine problems into something implicative of a solution. In the case of the knowledge base there is little support for this kind of iterative development of the understanding of a problem by lay users. 5) Customers are likely to vary in their technical ability, yet they must all use the same interface. In that case the terminology and information display are likely to prove difficult for novice users and tedious for experts. 6) To use the knowledge base it is necessary both to be able to interrogate it *and* to interpret the results. As it stands results are expressed technically and can require a significant knowledge of devices to interpret them.

All of this underscores the fact that systems using pure keyword-based searches like this one are not truly sufficient to provide the kinds of online support many users will require. However, as we indicated at the outset, heavily structured systems run the risk of proving too rigid, as well as being expensive to construct. In the light of this we have turned our own attention to working on retaining the flexibility of a keyword-based engine, whilst creating a user interface that can properly take into account the interactional components we have discussed. This will be accomplished through means of *in situ* linguistic analysis of user input that is coupled with a linguistic analysis of the contents of the online knowledge base. Amongst other things this provides for things like the reconciliation of symptomatic and technical terminologies, the addition of symptoms, and the iterative refinement of the description of a problem. Thus it can be seen that hybrid systems do not by any means always necessitate going back to the drawing board. Instead there are ways to devise solutions that can stand on top of existing knowledge bases, something that in its own right would seem to merit further research.

Acknowledgements

We thank members of the field work site for their patience, cooperation and help.

References

1. Aha, D. W., Maney, T., and Breslow, L. A. (1998) Supporting Dialogue Inferencing in Conversational Case-Based Reasoning. *Proceedings of EWCBR '98: Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning*, 1998. 262-273.
2. Anderson, R. J., (1994), Representations and Requirements: The Value of Ethnography in System Design. *Human-Computer Interaction*, Vol. 9. 151-182.
3. Bobrow, D. G., and Whalen, J. (2002) Community Knowledge Sharing in Practice: The Eureka Story, Reflections. *Journal of the Society for Organizational Learning*, Vol. 4 Issue 2.
4. Hughes, J, King, V, Rodden, T and Andersen, H, (1994) Moving Out from the Control Room: Ethnography in System Design, in R Furuta and C Neuwirth (eds), *Proceedings of the Conference on Computer Supported Cooperative Work*, October 22-26, 1994. 429-440

5. IDC Report #28967 (2003) *Worldwide and U.S. Software Support Services Forecast and Analysis, 2002-2007*, IDC Publications.
6. Jensen, F. V., Skaanning, C. and Kjærulff, U. (2001) The SACSO System for Troubleshooting of Printing Systems. *Proc. of SCAI 2001*. 67-79.
7. Kang, B. H., Yoshida, K., Motoda H., and Compton, P. (1997) Help Desk System. *Applied Artificial Intelligence*, 1 December 1997, Vol. 11, No. 7, 611-631(21).
8. Kolsky, E. (2004) *MarketScope: E-Service Suites*, 27 February 2004, Gartner Research
9. Ragsdale, J. (2005) Trends 2005: Customer Service And eService. *IT View and Business View Trends*, November 1, 2004, Forrester Research
10. Whalen, J. (1995) Expert systems versus systems for experts: computer-aided dispatch as a support system in real-world environments. *The social and interactional dimensions of human-computer interfaces*. 161--183, 1995.