



Conversational Help Desk: Vague Callers and Context Switch

Osamuyimen Stewart, Juan Huerta, Ee- Ee Jan, Cheng Wu, Xiang Li, David Lubensky

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA

[ostewart, huerta, ejan, chengwu, xiangli, davidlu]@us.ibm.com

Abstract

Two salient properties of user behavior make Help Desk a unique speech application different from the more general transactional kind: (a) majority of users have only vague ideas about their problem, and (b) these users are likely to context-switch (change discourse topic) during the course of a dialog. We describe a conversational Help Desk natural language call routing application and show how the alignment of Voice User Interface (VUI), Grammar Development, and Application Architecture results in a conversational user interface that is able to guide the vague user in the most optimal way while being flexible to allow mid-discourse context switches. Usability evaluation confirms the peculiar user behavior and provides empirical evidence that user’s perception of time in a speech application can be influenced by the dialog; in this case, Help Desk users’ tend not to become impatient going through three-five dialog turns, as long as dialog is progressing toward problem-resolution.

Index Terms: Spoken dialog management, voice user interface design, natural language understanding, technical support problem resolution, discourse analysis

1. Introduction

It is generally assumed that call routing via *natural language* is better suited for speech applications containing a large (usually more than 10, above 100 in many cases) number of menu options. In this regard, natural language refers to applications that have an initial open-ended prompt where users are not directed in terms of what they can or cannot say [1].¹ In response to the prompt, users may freely describe their request in their own words or utterances. We implemented the natural language solution for the IBM Help Desk which employees call for technical support on a variety of issues, and get routed to designated Agent skills group. There are 70 of such call routing destinations and each destination-bucket contains between 3 to 10 sub-distinctions such as hardware or software, type of machine or system, year of make, custom features, etc. Table 1 presents an analysis of 560 trouble tickets focusing on the first turn utterances of how users describe their problems to the human Agents:

¹ Sheeder and Balogh [2] have proposed a primed version of the open-ended prompt. For example, “Welcome to IBM Wireless. You can say “account balance,” “reset password,” “make payment,” or “customer service” What may I help you with?” We will not evaluate either of these versions in this paper.

Table 1. Breakdown of first turn user utterances

Technical Caller (clear description)	13%
Somewhat-technical caller (reasonable description)	17%
Non-technical caller (vague description)	70%

Table 1 summarizes our observation that users come to the Help Desk with very vague descriptions of their problem such as the name of the product. Thus, it provides empirical evidence that the Help Desk is a unique application that is different from the more pervasive transactional kinds of speech application. For example, transactional applications for banking, airline reservation, pharmaceutical ordering, etc. typically involve well-defined form-filling procedures where users can either “buy,” “sell,” or “retrieve” information. In these situations, majority of the users *know* what they want to buy or sell, and in the specific example of airline reservation applications, users come with reasonable information about city, date, or time, *ceteris paribus*. Blanchard and Stewart [3] have observed that users of natural language systems may sometimes say vague utterances. They describe specific prompting strategies as general steps in error-recovery and not as a defining characteristic of user behavior in transactional applications. Therefore, the major challenge in implementing a Help Desk application is the observation that 70% of users do not have good technical knowledge of the problem they have called about. In other words, majority of users are not able to exactly state the reason for their call at the first turn. Furthermore, our analysis reveals that one consequent characteristic of vague callers is that they very rapidly switch topic or refine their request from one domain to a different one during the course of a single interaction. The following dialog is a typical illustration of the user interaction:

System: Welcome to the IBM Help Desk. I am an automated assistant here to direct your call. How may I help you?

User: “I need support with my machine”

System: Ok. What is the name of the system or application?



User: “my password, I am locked out of my machine”

System: Alright, password support. Please say the name of the system you need help with?

User: “I need to speak with the support people in the Austin Labs”

System: Ok. I can help you with that. Please hold while I connect you with Austin Labs Help Desk.

The user begins the interaction with a general description of their problem, i.e., in response to the initial open-ended prompt, the user offers a very vague description of their problem by identifying the entity affected (my machine) but, it is not clear if this is about a hardware or software problem, or the kind of hardware or software. This triggers two dialog turns clarifying the *exact* intent. However, at the third turn, it becomes clear to the user that their *exact* intent is actually a request to be transferred to the Austin Labs Help Desk—resulting in a context switch whereby the user changes the dialog context from *password* to *location*. As section 2.1 will show, “Austin Labs” (a third turn request) is a specific class which the system could route on (at the first turn), since no further clarification is needed. However, characteristic of vague users, the *exact* intent only comes after multiple dialog turns.

2. The implementation

We now describe the Voice User Interface (VUI) design, Grammar Development, and Application Architecture decisions and strategies used to successfully handle vague responses and sudden context switches in the implementation of the IBM Help Desk call routing application.

2.1. Voice user interface (VUI) template

The initial task in the Voice User Interface (VUI) design was to determine the routing criteria, i.e., what constitutes sufficient and necessary conditions for routing a call? For the majority of requests, there are two things the caller must specify before they can be transferred to the correct Help destination:

- (a) The problem domain, e.g., password
- (b) The executable action (e.g., delete or reset) or the name of specific entity (e.g. Lotus Notes or VM System)

Thus, an optimal design of the VUI must distinguish vague utterances, e.g., “I need support with my machine” from those that are routable (containing sufficient and necessary information) like “I want to reset my Lotus Notes password.” Consequently, the VUI template is divided into three modules:

- Clear target: these are routable utterances defined as containing sufficient and necessary information for routing a call. For example “I want to change my antivirus setting”
- Vague target: these are non-routable utterances defined as containing one (or insufficient) piece of information for call routing. Two kinds of vague target are distinguished:
 - (a) Vague: this applies specifically to those utterances with only one piece of routing information (for example, a user says “Lotus Notes”) and the utterances does not specify the executable action (what is exactly wrong with

Lotus Notes or what they want done). (b) Opaque: this applies to utterances that are semantically relevant but too general to identify a problem domain (for example, a user says “I want support” or “my application just died”); both utterances fail to meet the routing criteria because many questions are left unanswered such as the kind of support or the name of machine they need support with etc.).

- Not information target: these are utterances that provide no relevant information whatsoever for routing a call. For example, a user says “Is this the Help Desk?”

Figure 1 is a representation of the mental model that combines these three modules into the Help Desk call routing template.

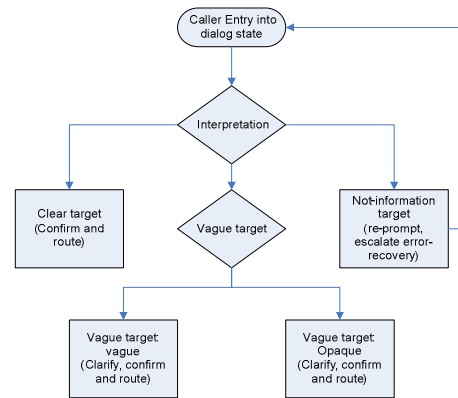


Figure 1 Help Desk call routing VUI template.

According to Figure 1, utterances that have the interpretation of clear target are routed to their desired destination (after confirmation), while those that belong to the not-information class receive standard error recovery handling involving re-prompt and escalated error prompts. Essentially, these callers return to the top of the template and start over again. The vague target is a more complex module and it is broken down into two sub-categories of non-technical callers as discussed in 2.1.1 below. However, when a positive return from one of the sub-categories completes the information for call routing, it becomes a clear target, and from there the call is routed to the right destination. Therefore, the mental model in Figure 1 sufficiently maps out the entire discourse domain of the Help Desk: an utterance can belong to one of three buckets or targets (clear, vague, and not-information).

2.1.1. Vague target, confirm, and context switch

We turn now to the residual issue of how to handle mid-discourse shifts commonly associated with vague users. Based on the discussion of Figure 1, the relevant places where the caller can change the context of a dialog thread are the vague targets and the confirm paths. In our template approach to VUI, we hypothesize that both vague and confirm functions share similar underlying structure. Thus, a single VUI template is proposed for handling clarification, confirmation, and context switches as shown in Figure 2.

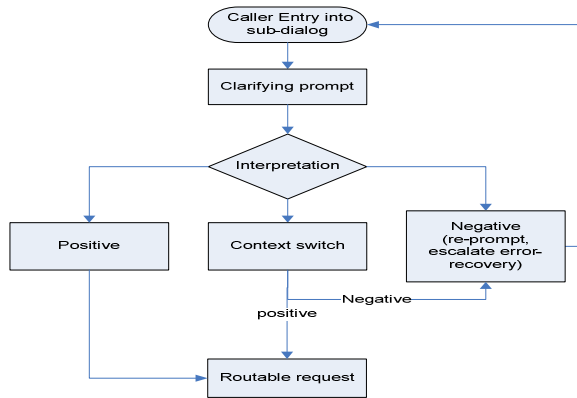


Figure 2 Template for vague, confirm, context switch.

According to Figure 2, there are three possible interpretations that can be derived after a clarifying/confirmation prompt: (a) a “yes” response (positive), (b) a context switch (new request), or (c) a “no” or out of grammar response. All positive values i.e., “yes” response or context switch (new request) result in a routable request, while negative responses (“no” or out of grammar) are passed on following standard error recovery procedures.

2.2. Grammar development

We leveraged Augmented Backus-Naur Form (ABNF) grammars and Natural Language Understanding (NLU) statistical language models. We trained an action classifier and the language model based on a corpus consisting of elicited sentences and transcription of actual Help Desk conversations.

In order to implement the natural language understanding component for the strategies summarized in Figure 1, a single action classifier was used for the entire application. The action classifier is made available at the initial caller entry. The automatic speech recognition (ASR) text input of the user’s response to the initial open-ended prompt is passed to the action classifier for intent determination. This is captured by the INTERPRETATION diamond in Figure 1. Accordingly, a user’s utterance may be categorized into one of the three major buckets or targets (with the associated semantic interpretation class): clear, vague, or not-information. This can be illustrated using the sample dialog in section 1. For that interaction, the following classes are returned by the action classifier:

- User’s 1st turn:** “I need support with my machine”
Action classifier: Request for support with a machine (vague)
- User’s 2nd turn:** “my password, I am locked out of my machine”
Action classifier: Request Password support (vague)
- User’s 3rd turn:** “I need to speak with the support people in the Austin Labs”
Action classifier: Request Location; Austin Labs (clear)

The output of each action classifier result is used for making call handling decision from the three targets: clear, vague, and not-information. Thus, the class associated with the third turn

utterance contains a specification of the problem domain and the entity or specific action desired within that domain—satisfying the necessary and sufficient conditions for call routing, i.e., a clear target.

For the VUI template in Figure 2 specifying the sub-dialogs associated with vague targets, confirmation and context switch, a systematic combination of ABNF grammars and action classifier is used. Each vague or opaque category is associated with an ABNF grammar containing the likely responses or choices that are available to the caller for that particular class. For example, when a caller says “Lotus Notes” a Lotus Notes vague ABNF grammar is activated. Thus, a user’s utterance in response to the clarifying prompt (vague, opaque, or confirmation) is first passed to the associated ABNF grammar for a match. When there is a match, the call proceeds to a clear target page. However, when there is no match in the ABNF grammar, which is typically due to context switch, then there is an automatic default to the action classifier where the result is either positive (context switch) or negative (out of grammar). In this manner, each vague target contains two contiguous understanding components: a salient ABNF grammar specific to the vague domain, and the action classifier which spans the entire application’s semantic domain.

Based on this approach, we are able to achieve high recognition accuracy when the item is within the smaller ABNF grammar, and also able to accommodate context switches by automatically and systematically defaulting to the action classifier that contains a larger corpus.

2.3. Application Architecture

In order to implement the architecture of the templates described in section 2.1, we designed and built the Help Desk application in a flexible and extensible way to support the interaction patterns needed to handle vague callers and accommodate context switch. We built the application in a JSP-based J2EE architecture producing VoiceXML markup (similar to the general commercial architecture described in [4]). The application is an MVC (model view controller) type 2, in which the Model is implemented using Jakarta Struts, the View are VoiceXML JSP pages, and the model is implemented using plain java objects.

In the presentation layer (implemented in VoiceXML), most pages refer to a Statistical Language Model grammar. This SLM grammar is available in all the JSP pages and allows the user to speak naturally. Additionally, each vague page might have one or more conventional ABNF grammars in parallel. When the user speaks, if an utterance is obtained via one of the active ABNF grammars, then the utterance and its annotation are sent back. Otherwise, the utterance obtained will be submitted back to the application controller, but the annotation field will be empty. If the destination target in the controller identifies a non-empty annotation field, it will proceed normally; otherwise it will invoke the semantic interpretation engine using the obtained utterance.

The semantic interpretation engine is based on shallow parsing call routing technology (similar to that described in [5]) and is responsible for extracting the meaning of the utterance. After the semantic meaning is obtained the application continues following the patterns of figure 1 and figure 2. The



termination (goal) is to get to a clear target and after a new target page is found the page is invoked and the interaction is carried out in similar fashion as described above.

3. Usability evaluation and discussion

One consequence of vague utterances and context switches is that opaque utterances may require three to five dialog turns before determining a user’s intent. This appears to be a peculiar trait of the Help Desk application since VUI best practices for transactional applications suggest a maximum of three dialog turns to avoid caller frustration (as described in [6], [7], etc). To verify this observation, we set up a usability evaluation involving 37 participants (2/3 were female and the remaining 1/3 were male). To ensure an equal distribution of technical and non-technical callers, we only used responses from 20 participants (ten of each). The task was the same for all participants, they were asked to make 5 scripted test calls to the Help Desk natural language call routing application and go through vague scenarios involving three to five dialog turns before the call is routed. They were requested to complete their testing within an hour. At the end of the test calls, they were asked to fill out a usability questionnaire containing eight questions. In response to two of the questions that are relevant for this paper, participants were asked: (a) did the system successfully recognize all your utterances? (b) “Did each test case take too long?” For the purpose of our analysis, we made sure that the 20 respondents answered “yes” to (a). The result of responses to question (b) is summarized in Table 2.

Table 2. Summary of responses

Non-technical Caller (“No” response)	95%
Non-technical Caller (“yes” response)	5%
Technical caller (“no” response)	68%
Technical caller (“yes” response)	32%

As summarized in Table 2, majority of users who call the Help Desk application expect to answer a lot of questions. It is important to note that most of the 32% technical users who stated that the test cases took too long also were diligent to note in the “comment” section of the questionnaire that the scripts were too simple. Thus, based on their comments, we infer that the technical 32% “yes” respondents were reacting to the fact that they would not say the vague utterances used in the test cases, to begin with, and not necessarily contradicting the generalization that majority of callers offer vague description of the problem they have called about. From a design and human factors perspective, the overall generalization from this is that users of the Help Desk application do not mind going through multiple dialog turns between three and five. This is consistent with similar conclusion by Boyce [8] who showed that users did not seem to mind the length of a prompt if the voice was pleasant.

Table 2 lends credence to a possible generalization that the psychology of the Help Desk application is different from other kinds of speech applications—users who call the Help Desk do

so because they have a problem (something is broken), and they can vaguely describe the technical details. Consequently, they have the tendency to change the context of the discourse as they characteristically go through multiple dialog turns.

4. Conclusions

The Help Desk is a unique type of speech application because majority of the callers have very vague technical knowledge of the problem they have called about. These callers are also very likely to say an entirely new request mid-discourse (context switch) as the thoughts become clearer. To solve this problem, we have presented a template approach to VUI. This is a process where a single design can be generalized over various functions. Thus, problematic discourse issues involving vague utterances, confirmation, and context switch receive a straightforward handling using the same design template. Furthermore, it is argued that the proper alignment of VUI, Grammar Development and Application Architecture produces an optimal conversational speech interface to handle vague callers as well as the associated behavior of mid-discourse context switching.

5. References

- [1] Gorin, A. Parker, B., Sachs, P., & Wilpon, J. “How may I help you?” Proc. of Interactive Voice Technology for Telecommunications Applications. IEEE Piscataway, NJ, (1996), 57-60.
- [2] Sheeder, T., and Balogh, J. “Say it like you mean it: Priming for structure in caller responses to a spoken dialog system” International Journal of Speech Technology, 6 (3), (2003), 103-111
- [3] Blanchard, H.E., Stewart, O.T., “Conversational re-prompting in natural language dialog” Proceedings of Annual Meeting of the Human Factors and Ergonomics Society. New Orleans, September 20-24, 2004, pp 708-711
- [4] Pieraccini, R. and Huerta J., “Where do we go from here? Research and Commercial Spoken Dialog Systems”, in Proceedings of 6th Sigdial Workshop on Discourse and Dialog, Lisbon, Portugal 2006.
- [5] Wu Ch., Lubensky, D., Huerta J., Li X., and Kuo H. K. "A Framework for Large Scalable Natural Language Call Routing Systems" 2003 IEEE international conference on Natural Language Processing and Knowledge Engineering Oct 26-29, 2003, Beijing, China.
- [6] Cohen, M., Giangola, J., Balogh, J., Voice User Interface Design. Addison Wesley, Boston. 2004.
- [7] Kotelly, B., The Art and Business of Speech Recognition: Creating the Noble Voice. Addison-Wesley, Boston, 2003.
- [8] Boyce, S. J. “Spoken natural dialog systems: User interface issues for the future” In D. Gardner-Bonneau (Ed.) Human Factors and Voice Interactive Systems. Kluwer, Norwell, MA (1999), 37-61.