Optimizing the Pause Length before Presentation of Always-Active Navigation Commands

TR 29.3594
November 14, 2002

Patrick M. Commarford
James R. Lewis

IBM Pervasive Computing

Boca Raton, Florida

Abstract

It is important to help users navigate speech-based interactive voice response systems (IVRs) by presenting the always-active navigation commands at any potential task end point. This paper presents an analysis we conducted to determine the optimal silence pause between presentation of a menu at a task terminal point and the presentation of the always-active navigation commands. Based on the distribution of response times to menu selection lists as a function of whether a desired option is in the list and whether the menu has been previously presented, we recommend a 1500 msec pause before presenting the navigation commands.

ITIRC Keywords

IVR
Speech interface
Telephony
VRU

Contents

Introduction

Interactive voice response (IVR) systems allow businesses to offload some of the costs of human labor for customer service activities. This is done by allocating to the system many of the tasks previously performed by human operators. Speech-based IVRs allow users to communicate with the system through speech, which can speed the interaction and reduce the total time per call. With speech-based interfaces, users do not need to remove the phone from their ears to use the system. In addition, the system often does not need to provide users with a set of options before they can respond, nor does it need to assign options to keypad presses.

It is important that the voice interface provide an easy and intuitive method for users to do tasks and navigate the system. One way to help users navigate the system is to present the always-active navigation commands at any place in the call flow that could be the end of a task (i.e., terminal point). After a user completes a task (such as a banking transaction), the system will typically prompt the user with a set of menu selection items (e.g., to review the transaction, to make another transaction, etc.). Since this is a potential terminal point (i.e., the user wanted only to make the transaction and then navigate to a different area of the interface), it is important to provide the navigation commands here. However, if this information plays immediately after the end of the menu selection list, usability problems can occur. Specifically, many of those users who do, in fact, want to make a selection from the menu will feel obligated to wait for the system to quit speaking before they make a selection. Alternatively, the system might interrupt users who have started to speak a desired option, causing them to stop and restart their command, which confuses the users and the speech recognizer (see Balentine & Morgan, 2001, for a discussion of the 'stuttering effect' and other usability issues related to barge-in).

To avoid these problems, it is important to provide an ample amount of silence before the presentation of the navigation commands. This allows users to make a selection from the list without interrupting or being interrupted by the system. On the other hand, the period of silence should not be so long that users give up and simply select an item from the list because they feel that they have no other options (a behavior observed in usability testing that also has the effect of suppressing the system from playing help messages). Another possible problem associated with a long delay before revealing the always-active navigation commands is that users will sometimes guess incorrectly at the appropriate navigation command. This will often trigger a first-level help message, which is commonly a more detailed description of the function provided by each of the unwanted menu items rather than a list of navigation commands.

The purpose of this study was to gather information that would help us to determine the optimal period of silence to provide between the end of a menu selection list and the beginning of the presentation of always-active navigation information at task terminal points. Specifically, we set out to determine the following:

1. How long users typically take to pick a menu option from a menu that clearly contains the desired option.

2. How long users typically wait to speak if no such option is presented.

3. Whether the above mentioned latencies are dependent on whether users have heard the specific menu before and are already familiar with the commands.

4. The mean times to respond to a menu as a function of menu length.

Method

Participants

We analyzed video segments from two speech-based IVR usability studies. Participants were six individuals recruited from a temporary employment agency in the area. Four males (2 < 40 years old; 2 > 40 years old) and two females (> 40 years old) participated in the studies.

Materials

We used a digital video camera to capture usability data in the studies. The digital video was converted to a set of .mpeg files with Pinnacle[1] Studio 7 editing software before analysis. For one experiment, participants used a phone-like interface in conjunction with a VXML prototype of a voice user interface, running on an IBM ThinkPad[2]. For the other experiment, participants used an actual telephone in conjunction with a prototype voice user interface running on a speech server. A stopwatch was used to measure response times.

Procedure

We analyzed the video files and recorded response latencies for instances in which the following occurred:

1. The system presented the user with a menu that included a selection that clearly met the user's immediate goal.

2. The system presented the user with a menu that clearly did not include a selection that met the user's immediate goal, and the user did not issue a valid, goal-driven command.

We defined response latency as the amount of time that elapsed from the end of the last menu option to the start of the user response. If a user interrupted the menu to make a selection before the menu had been presented in its entirety, we set the time to 0 seconds. We recorded three measurements of each response time by hand timing with a stopwatch while viewing the video files. We also took note of: a) whether the user had previously heard the specific menu and b) the number of menu items in the list. We did not include latency data from any instances that did not clearly fall within one of the specifications above (for example, if a useful option was in the selection list, but the user did not realize that the option would be helpful). Also, we did not record latencies when users responded with a valid, goal-driven command that was not in the menu. For the "not in menu" data, we included only instances in which users made menu selections that clearly were not in-line with their goals or tried some other out-of-grammar command.

[1] Pinnacle is a registered trademark of Pinnacle Systems, Inc.
[2] IBM and ThinkPad are registered trademarks of IBM, Corp.

Results

Effect of Menu Length

Table 1 lists the mean response times, in seconds, based on total number of items in the menu. The table shows no clear trends in the amount of time to respond as a function of menu length.

Table 1. Mean response times (in seconds) for each number of menu items

Menu Items	N	Mean Response Time	Standard Deviation
2	27	1.79	2.03
3	51	2.70	6.84
4	21	2.06	3.71
5	15	1.74	1.82
6	3	0.43	0.74
9	10	2.76	3.05
10	19	0.55	1.10
Overall	146	2.02	4.51

Effect of Presence of Desired Option in List and Previous Experience with Menu

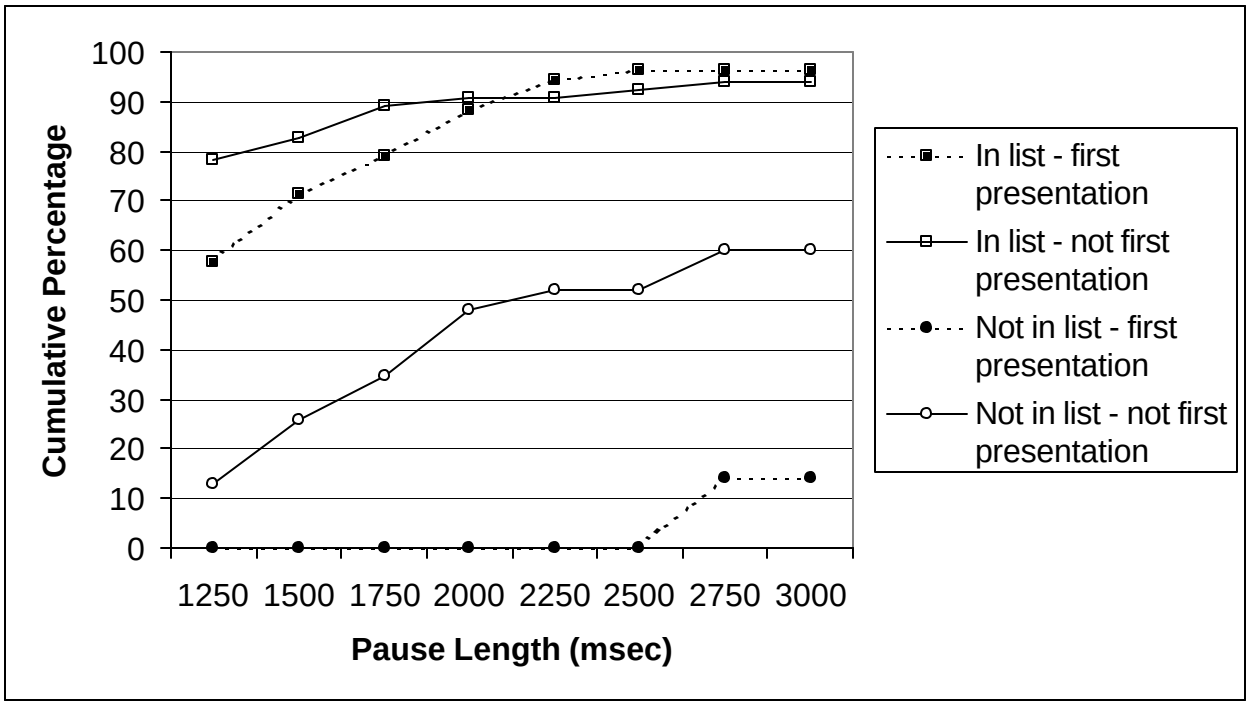
Table 2 lists the percentage of user responses in each condition started within 1250, 1500, 1750, 2000, 2250, 2500, 2750, and 3000 msecs. Table 2 also indicates whether the desired menu option was in the list and whether it was the first presentation of the menu (indicated by "1st pres?"). The table shows that when a desirable option was in the selection list for the initial presentation of the menu, users started a selection within 2500 msecs 96.2% of the time and made a selection within 1250 msecs 57.7% of the time. Users made no selections in the time period ranging from 2500-3000 msecs after the end of the menu presentation. When users had previously heard the menu items, they typically responded a little more quickly, with 78.1% of responses starting within the first 1250 msecs.

Table 2 also reveals that in the seven clear cases in which no desirable selections appeared in a menu presented for the first time, users never began a response within the first 2500 msecs following the last menu item presentation. However, when subsequently presented with the menu, (including those instances in which the first presentation *did* include an option that met user needs), users tried an unhelpful command 52.2% of the time within 2500 msecs. These data also show that users typically responded more quickly to menus with which they were familiar (as indicated by the percentage of responses given in less than 1250, 1500, and 1750 msecs). The final row gives the user response time data, regardless of whether a good option was in the list and regardless or whether the menu had not yet been presented. Figure 1 graphically illustrates the response time patterns of the four studied conditions.

Table 2. Percentage of user responses started within 1250-3000 msecs

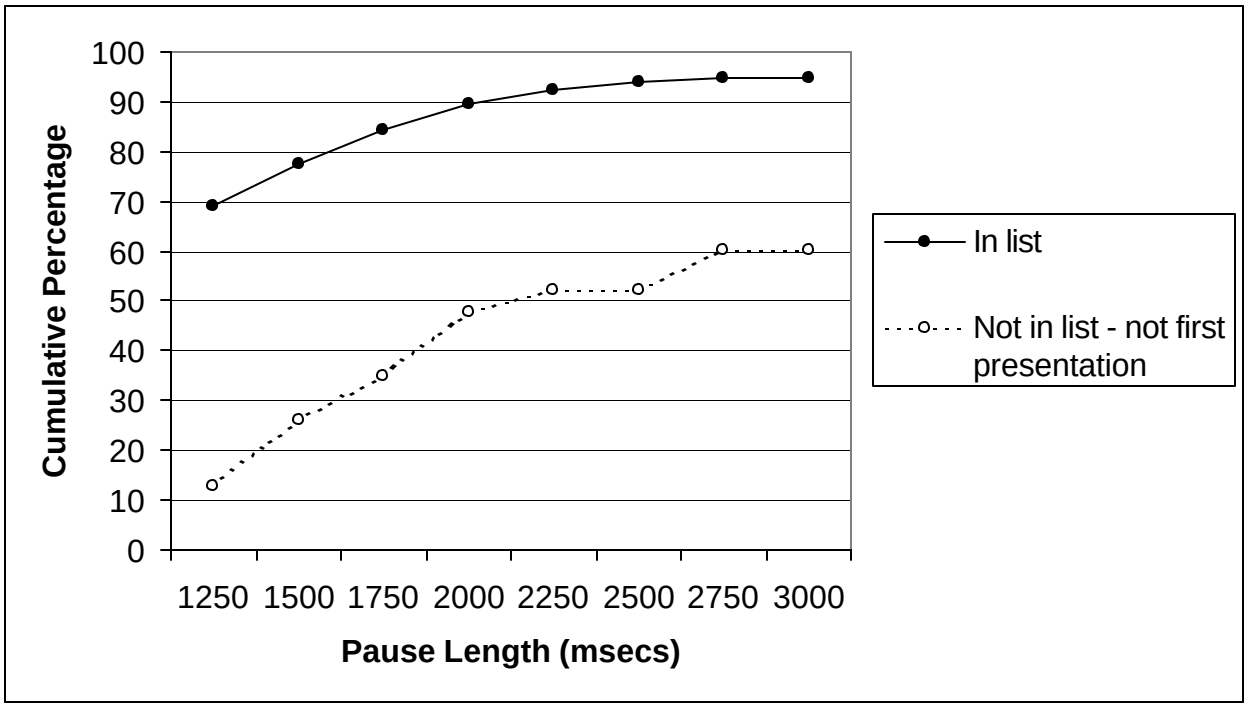
Option in list?	1st pres?	N	1250	1500	1750	2000	2250	2500	2750	3000
Yes	Yes	52	57.7	71.2	78.9	88.4	94.2	96.2	96.2	96.2
Yes	No	64	78.1	82.8	89.1	90.7	90.7	92.2	93.8	93.8
Yes	Overall	116	69.0	77.6	84.4	89.7	92.2	94.0	94.8	94.8
No	Yes	7	0	0	0	0	0	0	14.3	14.3
No	No	23	13.0	26.0	34.8	47.8	52.2	52.2	60.1	60.1
No	Overall	30	10.0	20.0	26.7	36.7	40.0	40.0	50.0	50.0
Overall	Yes	59	50.1	62.7	69.5	78.0	83.1	84.8	86.4	86.4
Overall	No	87	60.9	67.8	74.7	79.3	80.34	81.6	85.1	85.1
Overall	Overall	146	56.7	65.8	72.3	78.8	81.5	82.9	85.6	85.6

Figure 1. Cumulative percentage of response times as a function of condition



The distribution of response times for the two 'in list' conditions were similar (and virtually identical from 2000 msecs on). The distribution of response times for the two 'not in list' conditions, on the other hand, were markedly different. For first presentations with the option not in the list, the fastest response occurred more than 2500 msecs after the end of the menu. For subsequent presentations with the option not in the list, about 50% of responses occurred before 2500 msecs. This suggests that the optimization problem is to achieve a balance between the distributions of the 'in list' conditions (combining initial and subsequent presentations) and the distribution of the 'not in list – not first presentation' condition, as shown in Figure 2.

Figure 2. Revised presentation of cumulative percentage of response times



Discussion and Recommendations

Menu length does not appear to be an important determinant of response time. The factors of in-list versus not-in-list and initial versus subsequent exposure to the menu do appear to be important. Referring back to Figure 2, the optimization problem is to maximize the cumulative percentage for the in-list condition while simultaneously minimizing the cumulative percentage for the not-in-list condition. The in-list distribution is monotonic and negatively accelerated with relatively little increase in the cumulative percentage past 2000 msecs. The not-in-list distribution starts at a much lower level (13% at 1250 msecs), but increases rapidly through 2000 msecs, where it is just under 50%. This sets the boundaries for the optimal pause between 1250 and 2000 msecs.

However, there are two issues that complicate the solution of the optimization problem: the relative severity of the usability problems associated with failing to optimize the in-list percentage versus failing to optimize the not-in-list percentage and uncertainty about the extent to which the not-in-list distribution would change if the users had heard the list of always-active navigation commands in the initial presentation of the menu.

Relative Severity of Usability Consequences Due to Failure to Optimize

It seems reasonable that not hearing the always-active navigation commands when a user needs them will create more severe usability problems than a potential interruption by this command list when a user does not need it. Here's a typical situation:

> User: (Listens to a news story.)
> System: Select next, repeat, or previous.
> User: Next.
> System: (Plays next news story.)
> System: Select next, repeat, or previous.

For this second presentation of this particular menu, assume that the user does not want any of these options (e.g., wants to start over). About 50% of the time, users would jump in and say "Next," "Previous," or make some other out-of-grammar attempt to navigate before hearing the always-active navigation terms if the pause exceeded 2000 msecs, leading them to suffer the adverse usability consequences associated with not hearing the always-active commands when needed.

Specifically, when the presentation of the information is too late, users often select a menu item even though the item will not meet their immediate goals. They continue to select items and to work through menus in their attempt to navigate away from the menu. Sometimes they also try making up their own navigation commands, which often are not in grammar. This can lead to excessive time-on-task and a great deal of user frustration. On the other hand, if users hear the navigation information at a time when they do not need it, they might (a) wait for the prompt to end or (b) start to make a selection and then quit speaking as soon as the system starts. This can be irritating, but is less likely to affect task time and successful task completions. (Note that the likelihood of occurrence of interruption is greater with recognition-based barge-in than with energy-based barge-in, but this is only one of several issues to consider when choosing a barge-in method.)

Because the situation of failing to hear the navigation commands when users need them is more detrimental than interrupting users, the apparent optimal pause would be 1250 msecs (minimizing the cumulative percentage for the not-in-list distribution at the expense of failing to maximize the cumulative percentage for the in-list distribution).

Potential Effect of Providing Always-Active Navigation Commands on Not-In-List Distribution

Unfortunately, the "not-in-list subsequent presentation" data might only have the observed distribution when the system does not provide the always-active navigation after potential terminal points. Referring back to Figure 1, note that on the initial presentation of the menu for the not-in-list condition, the users waited a substantial amount of time before making a response (more than 2500 msecs). Had the users heard the always-active navigation commands at any time during that pause, they might well remember the appropriate command when needed (or would at least be aware that they could hear those commands by simply waiting a short time after the end of the basic menu). This suggests that the optimal pause is probably (but not definitely) longer than 1250 msecs.

Final Recommendation: Set the Pause to 1500 Msecs and See What Happens

Based on these analyses, it appears that 1500 msecs is a reasonable pause to use (satisfies the design tradeoffs to the best of our current knowledge). There is some uncertainty as to whether this is truly the optimal pause. As we conduct future usability studies on systems or prototypes with this pause set to 1500 msecs, we must continue to monitor user behavior to help us understand whether we need to adjust the pause length. It is likely, though, that a pause of 1500 msecs will work well.

References

Balentine, B., Morgan, D. M., & Meisel, W. S. (2001). *How to build a speech recognition application: A style guide for telephony dialogs* (2nd ed.). San Ramon, CA: Enterprise Integration Group.