2007

A Study of Out-of-turn Interaction in Menu-based, IVR, Voicemail Systems

Saverio Perugini
University of Dayton, sperugini1@udayton.edu

Taylor J. Anderson
University of Dayton

William F. Moroney
University of Dayton, wmoroney1@udayton.edu

eCommons Citation

Perugini, Saverio; Anderson, Taylor J.; and Moroney, William F., "A Study of Out-of-turn Interaction in Menu-based, IVR, Voicemail Systems" (2007). *Computer Science Faculty Publications*. 26.
https://ecommons.udayton.edu/cps_fac_pub/26

A Study of Out-of-turn Interaction in Menu-based, IVR, Voicemail Systems

Saverio Perugini
Dept. of Computer Science
University of Dayton
Dayton, OH 45469–2160
saverio@udayton.edu

Taylor J. Anderson
Dept. of Psychology
University of Dayton
Dayton, OH 45469–1430
anderstj@notes.udayton.edu

William F. Moroney
Dept. of Psychology
University of Dayton
Dayton, OH 45469–1430
moroney@udayton.edu

ABSTRACT

We present the first user study of out-of-turn interaction in menu-based, interactive voice-response systems. Out-of-turn interaction is a technique which empowers the user (unable to respond to the current prompt) to take the conversational initiative by supplying information that is currently unsolicited, but expected later in the dialog. The technique permits the user to circumvent any flows of navigation hardwired into the design and navigate the menus in a manner which reflects their model of the task. We conducted a laboratory experiment to measure the effect of the use of out-of-turn interaction on user performance and preference in a menu-based, voice interface to voicemail. Specifically, we compared two interfaces with the exact same hierarchical menu design: one with the capability of accepting out-of-turn utterances and one without this feature. The results indicate that out-of-turn interaction significantly reduces task completion time, improves usability, and is preferred to the baseline. This research studies an unexplored dimension of the design space for automated telephone services, namely the nature of user-addressable input (utterance) supplied (in-turn vs. out-of-turn), in contrast to more traditional dimensions such as input modality (touch-tone vs. text vs. voice) and style of interaction (menu-based vs. natural language).

Author Keywords

Out-of-turn interaction, Interactive Voice-Response systems (IVRs), Automated Telephone Services (ATS), speech user interfaces, user studies, Automatic Speech Recognition (ASR), mixed-initiative interaction, usability.

ACM Classification Keywords

H.5.2. [Information interfaces and presentation (e.g.,HCI)]: User interfaces – *interaction styles, voice I/O*.

INTRODUCTION

Automated, menu-based, interactive voice response systems (IVRs) are ubiquitous. They are frequently used in call centers and voice portals (*TellMe*), for directory assistance and voicemail (Siemens *PhoneMail*), and in the banking (Citibank) and airline (Delta) industries to reduce agent costs. While a menu design provides a familiar motif for users, it can also be a source of frustration when the structure of the menus does not match the user's model of the task. Specifically, users are faced with selecting a menu item at each level which best fits the task. This can often lead to dead-ends and back-tracking [18]. For instance, consider dialog 1 (simplified for purposes of presentation) between a user and a voicemail IVRs.

Dialog 1: directed

1 **System**: Welcome to voicemail. Main menu. Please say one of the following: 'listen,' 'send,' 'system settings,' or 'answering options.'
2 **User** (to self): Hmmm. I just want to set my password. Not sure which to select. I'll try listen.
3 **User**: Listen.
4 **System**: Okay listen. You have no new messages.
5 **System**: Main menu. Please say one of the following: 'listen,' 'send,' 'system settings,' or 'answering options.'
6 **User**: Answering options.
7 **System**: Okay answering options. Please say one of the following: 'personal greetings,' 'notification options,' or 'answer mode.'
8 **User** (to self) Password is personal, but it's not a greeting. Let me back out and try system settings.
9 **User**: Main menu.
10 **System**: Okay main menu. Please say one of the following: 'listen,' 'send,' 'system settings,' or 'answering options.'
11 **User**: System settings.
12 **System**: Okay system settings. Please say one of the following: 'groups,' 'password,' or 'additional settings.'
13 **User**: Password.
(dialog continues with set-password activity)

Here the user is interested in setting the password, but is unable to ascertain which of the four top-level menu choices will eventually lead to task completion. Her first choice — 'listen' (line 3) — leads to a dead-end. The user's second choice — 'answering options' (line 6) — leads to options

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.

which the user deems irrelevant to the task. Thus, the user backs out of this path by saying 'main menu' (line 9) and selects the 'system settings' item (line 11) which ultimately leads to successful completion of the task.

Notice that in this dialog the user must directly respond to the system prompts in the order in which they are played, and the system, on the other hand, is unable to deviate from its predefined script. Such a dialog is said to be a *directed dialog* and we say the user is making *in-turn* (or solicited) responses at each step. Due to the hardwired nature of the menu design, setting the password involves trail and error and potentially a taxing series of drill-downs and roll-ups to traverse the path through the menus leading to task completion. This problem is endemic to all menu-based systems.

Solution Approach

An approach to this problem is a technique we call *out-of-turn interaction*. The idea is to permit the user to make unsolicited utterances when unsure how best to respond to the current prompt. This technique is illustrated in dialog 2 (also simplified for purposes of presentation).

Dialog 2: mixed-initiative

1 **System**: Welcome to voicemail. Main menu.
Please say one of the following:
'listen,' 'send,' 'system settings,' or
'answering options.'
2 **User**: Password.
3 **System**: Okay password. Please say one of
the following: 'set' or 'remove.'
4 **User**: Set.
(dialog continues with set-password activity)

In contrast to dialog 1, rather than trying to predict which of the four top-level choices lead to the set-password facility, here the user says 'password' out-of-turn (line 2). This causes the dialog to immediately focus on the password submenu (line 3). At this point, the user decides to respond directly to the prompt and say 'set' (line 4). Progressive utterances are interpreted as a conjunction. The implicit assumption is that the out-of-turn utterance is not only relevant to the task at hand, but also a valid response to a forthcoming prompt. Interleaving out-of-turn utterances (line 2) with in-turn responses (line 4) has been recognized as a simple form of mixed-initiative interaction [1]. Therefore, dialog 2 is said to be a *mixed-initiative dialog*.

In this paper, we present a user study to evaluate the effectiveness and usability of out-of-turn interaction (vs. the baseline in-turn interaction) in a menu-based, IVRs to voicemail. Our results indicate that out-of-turn interaction reduces task completion time and is preferred. In exploring the nature of the user-addressable input (utterance) supplied (in-turn vs. out-of-turn), our study is fundamentally distinct from other research which has focused on more traditional design dimensions such as input modality (touch-tone vs. text vs. voice) or interaction style (menus vs. natural language) [13]. We first discuss the details of out-of-turn interaction including interpretations for it and survey related research wrt these design dimensions, and then discuss our comparative user study, its results, and contributions.

OUT-OF-TURN INTERACTION

What does it mean to interact out-of-turn?

There can be several interpretations of out-of-turn interaction, but only one should be used in an implementation for purposes of consistency [11]. Here, we assume that an out-of-turn utterance indicates that the user desires to experience a sequence of progressive steps through the menus where a subset of the terms in the utterance are involved in the menu choices within that sequence. We use the decision tree for Siemens (ROLM) *PhoneMail* voicemail system shown in Fig. 1, which is used in several organizations, as a running example to illustrate this operational interpretation.

Consider processing the out-of-turn utterance 'greeting' spoken from the home state of Fig. 1. Using the interpretation given above, we first retain each sequence through the menus which involve the term in the utterance as a menu item and prune out all others (similar to [18]). While there are 40 sequences in total from the home state of *PhoneMail* to each terminal item (those without further sub-menus), only the following five contain a menu item named 'greeting' and, therefore, would remain following the utterance 'greeting' spoken from the home state of *PhoneMail*:

≺answering options, personal greetings, change regular greeting, change (no answer) greeting≻,
≺answering options, personal greetings, change regular greeting, change (busy) greeting≻,
≺answering options, personal greetings, change alternate greeting≻,
≺answering options, personal greetings, select greeting, regular greeting≻,
≺answering options, personal greetings, select greeting, alternate greeting≻.

Fig. 2 (left) illustrates the pruned menu structure, which contains only the sequences above, resulting from speaking 'greeting' out-of-turn. It is important to note that the dialog structure is not flattened as a result of interacting out-of-turn, but rather preserved. Note also that while many menu items are removed from the entire menu structure, only those removed from the home state are salient to the user (until they start drilling-down).

There are a few practical, post-processing optimizations which we can conduct. Consider that, while dependent on the structure of the menus at the time that the out-of-turn utterance is spoken, an out-of-turn interaction often results in some menus with only one option. For example, in Fig. 2 (left), 'personal greetings' is the only menu item under the 'answering options' menu which, similarly, is the only menu item from the home state. A menu containing only one item implies that the item is no longer an option. In these cases, we follow classic menu-design research which indicates that the no menu should contain less than two items. Single-item menus should be consolidated with the previous menu (through which it is accessed) or the menu to which it leads. Fig. 2 (right) shows the final menu structure resulting from saying 'greeting' out-of-turn from the home state of *PhoneMail* and illustrates that the 'answering options' and 'personal greetings' items from Fig. 2 (left) have been removed.

Sometimes an out-of-turn interaction results in a single sequence. For example, saying 'password' from the home state of *PhoneMail* results only in the sequence: ≺mailbox options, password≻. In this case, the consolidation of single-

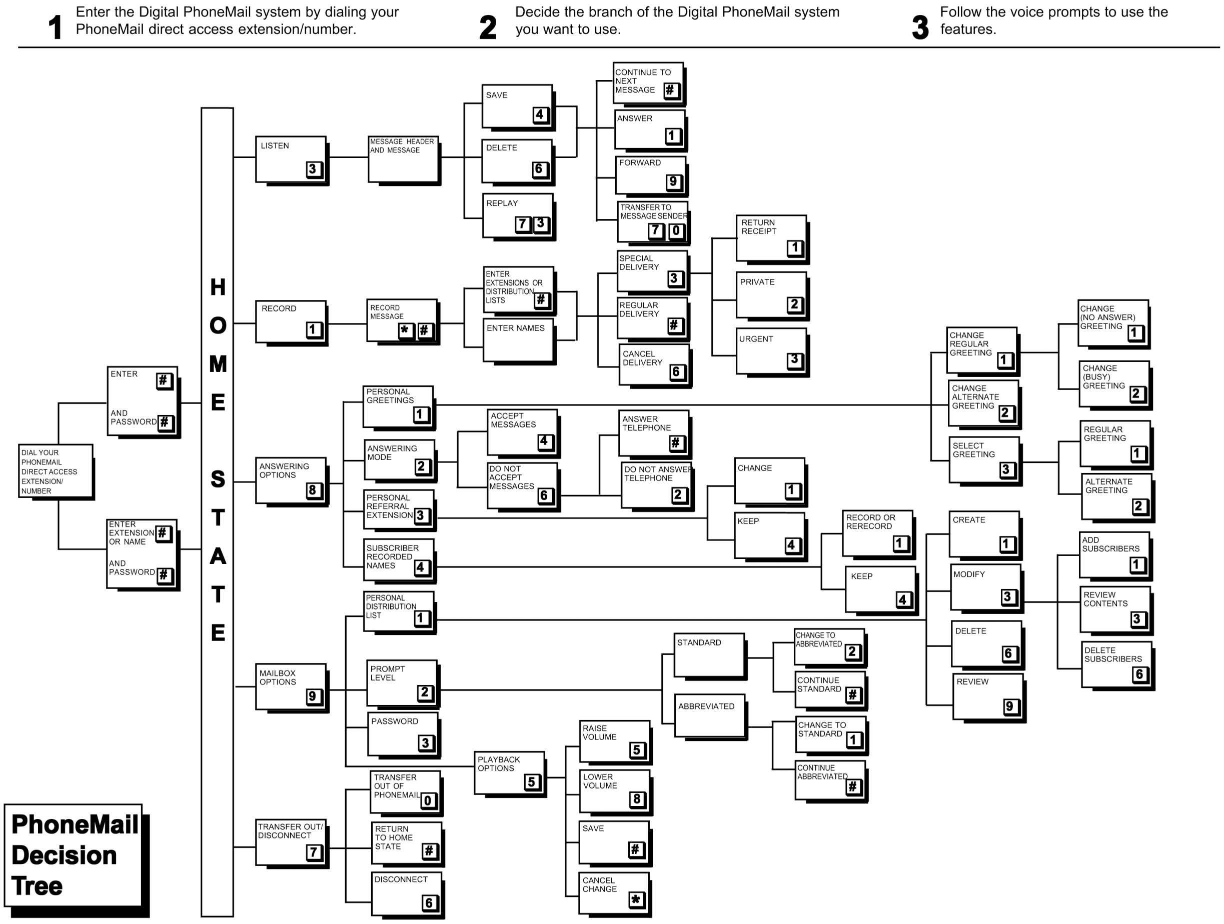

Figure 1. Series of hierarchical menus available in Siemens *PhoneMail* voicemail system.

item menus mentioned above results in a shortcut directly to the password facility without further prompting.

However, it is important to note that the main menu is not always among the single-item menus. For instance, consider saying ‘change’ from the home state in Fig. 1 which results in the menu structure shown in Fig. 3 (left). Now two choices remain from the home state because sub-menus containing items labeled with the term ‘change’ are nested under both the ‘answering options’ and ‘mailbox options’ menus. Notice further that in this example one menu choice in Fig. 3 (left) is labeled exactly with what was spoken out-of-turn (i.e., ‘change’). In such cases, since the user has effectively selected such items through the utterance (obviating the need for furthering prompting), we can remove any menu item (prompt) which exactly matches the out-of-turn utterance. However, we do not remove the facility accessed through it. Rather it is now accessed through the menu item predecessor of the removed item. Therefore, the change-personal-referral-extension facility is now accessed through the sequence ≺answering options, personal referral extension≻ (see Fig. 3, right).

Notice that while the organization of the menus resulting from an out-of-turn utterance is different in each example above, the interpretation of (processing) it, on the other hand, is fixed. Note also that the presentation of the menu prompts is never re-ordered. Prompts are are only pruned as a result of interacting out-of-turn. The original order of the remaining menu prompts is sustained. More importantly, the dynamic reduction of the tree, and thus the vocabulary, actually improves speech recognition accuracy, in contrast to de facto degradation of recognition accuracy common to most systems with support for mixed-initiative interaction.

In summary, out-of-turn interaction is optional and can be invoked (and interleaved with in-turn utterances) by the user at multiple points in a dialog at the user’s discretion. Moreover, future (in-turn or out-of-turn) utterances are cast within the context of past utterances. When the user speaks out-of-turn, we

1. retain each sequence through the menus which involves a subset of the term(s) in the utterance as a menu item and prune out all others,

2. remove the menu item(s) addressed by the utterance from each remaining sequence,

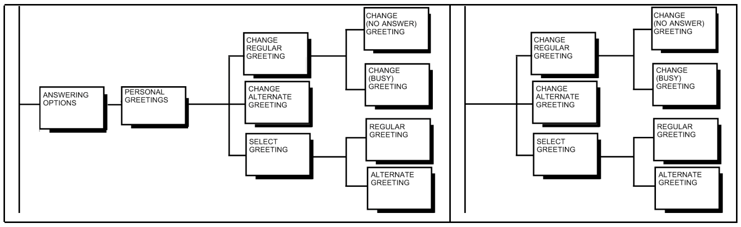

Figure 2. (left) Intermediate structure of menus resulting from saying 'greeting' out-of-turn from the home state of *PhoneMail*. (right) Final menu structure following from post-processing (left) to consolidate single-item menus.

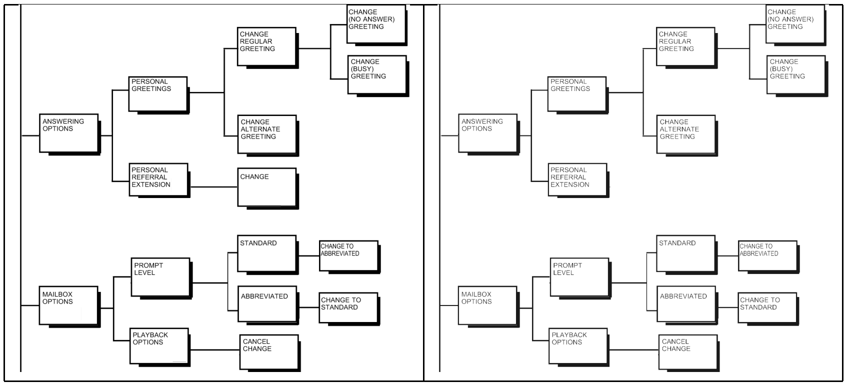

Figure 3. (left) Intermediate structure of menus resulting from saying 'change' out-of-turn from the home state of *PhoneMail*. (right) Final menu structure following from post-processing (left) to remove items exactly matching the out-of-turn utterance.

3. collapse any series of menus from the remaining organization, where each contains only one item, and
4. re-play the (reduced set of) prompts from the main menu.

In some cases, steps 2 and 3 may not remove any further choices from the menus remaining after step 1.

There can be several variations of this basic technique with minor differences in the interpretation and implementation details. However, the main idea is the same: permit the user to respond to a prompt nested deeper in the dialog structure before it is played, reducing the dialog appropriately.

Why interact out-of-turn?

Depending on the application domain, there can be several reasons for interacting out-of-turn. In the voicemail examples given here, out-of-turn interaction helps isolate the menu choice(s) relevant to the user's task by pruning out irrelevant options. The user is provided with auditory feedback when the new set of menu choices are played. We use the word 'isolate' rather than 'determine' (the menu choice) since the user may still need to predict which menu item will lead to task completion when out-of-turn interaction leaves the main menu with more than one choice. However, at this point, the user has the option of interacting out-of-turn again to hone in on the appropriate choice.

What out-of-turn interaction is not

An out-of-turn utterance does not involve natural language. By *natural language* we mean an IVRs which employs an open-ended prompt such as 'How may I help you?' [4]. Recall that the out-of-turn utterance is limited to a valid response to a prompt nested deeper in the menu organization; no other form of speech dialog is involved. Concomitantly, out-of-turn interaction also is not a hardwired menu. It is a hybrid between menu-based and natural language solutions. It is more flexible than fixed, hierarchical menus, but less open-ended than solutions involving natural language. Out-of-turn interaction is also not simply a search of the terminal objects (e.g., voicemail messages) themselves [16]. We shall have more to say about where our research is situated within the conceptual design space for automated telephone services (ATS) in our survey of related research below.

Out-of-turn interaction is not *barge-in* [6]. Barge-in permits the user to respond to the *current* prompt for input before it is played. Out-of-turn interaction, on the other hand, empowers the user to respond to *any* prompt nested deeper in the menu organization before it is played. Out-of-turn interaction and barge-in are orthogonal techniques and can be used in concert if desired, and are in our study. Lastly, while an out-of-turn interaction can result in a shortcut, (depending on the structure of the menus at the time of the out-of-turn utterance), it is not simply a shortcut to a menu nested deeper in the dialog. The shortcuts approach involves anticipating all points in the dialog where the user might desire to skip the current prompt and including mechanisms (e.g., a link) to transfer control to an alternate menu. On the other hand, out-of-turn interaction never augments the original menu structure.

How do users know what to say?

Knowing what to say is a problem endemic to all speech interfaces [11, 17]. Here, the only valid terms in an utterance are those used to describe the choices remaining in the menu at any time. This, of course, requires the user to be either fa-

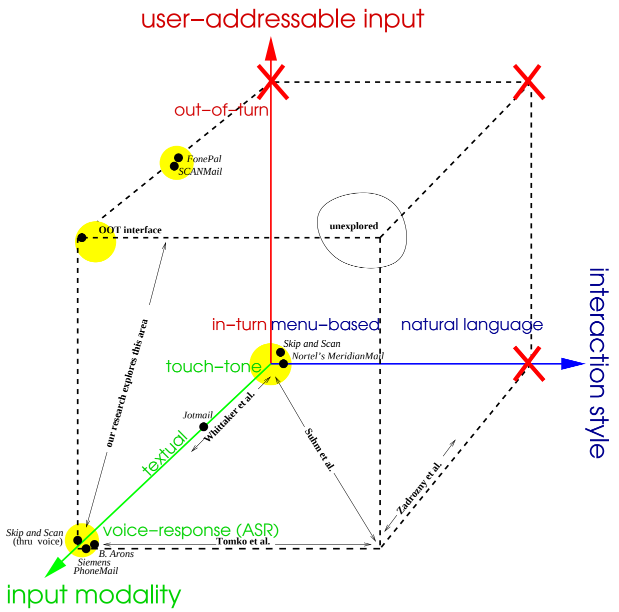

Figure 4. Conceptual design space for automated telephone services.

miliar with the terms used throughout the menu (in the case of a returning user) or to have a basic understanding of the general nomenclature of the underlying domain (in the case of a new user). This requirement may be optimistic, especially in esoteric or emerging domains where no standard nomenclature has been established. However, since a byproduct of the reduced menu is an improvement in speech recognition accuracy, expanding our lexicon with synonyms for menu items (e.g., passcode or PIN for password) is within the scope of viable approaches to this problem.

Related Research

There are several ways to study automated telephone services. The design dimensions which are most relevant to our research are the *nature of the user-addressable*[1] *input* (in-turn vs. out-of-turn), *input modality* (touch vs. text vs. voice), and *interaction style* (menu-based vs. natural language). Fig. 4 illustrates the conceptual design space described by these dimensions and situates related work within the space. Note that three corners — (menu-based, touch-tone, out-of-turn), (natural language, touch-tone, in-turn), and (natural language, touch-tone, out-of-turn) — of the cube are undefined. You cannot communicate out-of-turn information using a touch-tone modality. Similarly, you cannot use natural language through a touch-tone modality. While the majority of deployed systems in industry lie at the origin, over the past few years they have been shifting down the modality axis toward the voice-response end [12], e.g., Siemens *PhoneMail* is now available in both touch-tone and voice versions. Research has tended to focus on areas away from the origin.

Suhm *et al.* [13] explored the area between the (menu-based, touch-tone, in-turn) and (natural language, voice-response, in-turn) points and found that users are more frequently routed to the correct agent (and prefer) using systems residing at the latter point. Yin and Zhai [18] describe *FonePal*, a system which permits the user to search the decision tree of an IVRs like they would a hierarchical website. While the spirit of the search strings involved are out-of-turn (even though technically the system solicits the strings from the user through a search box), users enter the search strings using a textual modality which, unlike out-of-turn interaction, involves a context switch. Results indicate that users were faster on average at searching than browsing.

Resnick and Virzi [8] offer the *Skip and Scan* approach to designing and interacting with an ATS. *Skip and Scan* involves fragmenting each menu in an existing ATS decision tree into a series of several menus by enumerating purely navigational links between each to skip forward and back. The user then, when interacting with a system augmented with these links, can easily navigate back and forth through menus without first having to listen to all of the prompts for a particular menu. Notice that while *Skip and Scan* tackles *within-menu* navigation, we are studying *between-menu* navigation. Moreover, since the users are prompted to follow each of these purely navigation links, this approach also involves in-turn responses. While hardwiring additional navigation paths is one approach to increasing the scope of addressable information, support for out-of-turn interaction does not require augmenting the structure of the decision tree. Rather it requires transforming it (in real-time) during the processing of an out-of-turn input. Results indicate that users were faster with a *Skip and Scan* interface and preferred it to standard menus. Whittaker *et al.* [15] explored the area between the (menu-based, touch-tone, in-turn) and (menu-based, text, in-turn) points. Zadrozny *et al.* [19] explored the portion between the (natural language, voice-response, in-turn) and (natural language, text, in-turn) points. Arons [2] studied interacting with an ATS decision tree through speech input (front, bottom-left corner of Fig. 4) and Tomko *et al.* [14] studied the area between the (menu-based, voice-response, in-turn) and (natural language, voice-response, in-turn) points in the *Speech Graffiti* project. To the best of our knowledge, no study has explored the area between the (menu-based, voice-response, out-of-turn) and (menu-based, voice-response, in-turn) points and, therefore, we contribute a study which does. While most appropriate for casting our work, these three dimensions are not the only by which to design and study ATS (e.g., see [9]).

Examining the intrinsic nature of system-manipulated information reveals a dimension with a content- vs. structure-based dichotomy. Therefore, a different, but related, problem which has received attention is that of *browsing and searching* or *managing/prioritizing* the (terminal) voicemail messages themselves [10] as opposed to the *access* of them described here. For example, *FonePal* [18], out-of-turn interaction, and *Skip and Scan* [8] focus on customizing access (structure) to terminal objects, while Jotmail [15], *ScanMail* [16], and *TalkBack* [5] focus on searching/manipulating the terminal objects (content). Specifically, here we focus on non-serial interaction with a menu [9] rather than non-serial interaction with a terminal object itself. Though not the fo-

[1] By *addressable* information we mean the information which the system can accept from the user or, in other words, the information that the user can supply. We do not mean information that the system indexes (addresses).

cus of this paper, out-of-turn interaction can be adapted (with the help of segmentation) to work in content-based situations as well.

Out-of-turn interaction is not mutually-exclusive with any of these approaches. Rather, it affords different, but complementary, interaction to enumerative [8], visual [15], multimodal [18], or content-based [5] approaches. Out-of-turn interaction is a simple, optional, uni-device, uni-modality, transformation-based approach which does not involve any augmentation of the original phone tree, subsumes traditional interaction, and is applicable from day-one.

OUR STUDY

Objectives

The goal of our research was to evaluate the effect of interacting out-of-turn with a menu-based, IVRs on task completion time, usability, and preference. Since out-of-turn interaction automatically removes options from the menus which are not related to the utterance spoken — thus preventing the user from exploring irrelevant paths — we expected that it would increase the task success rate and reduce task completion time. We also expected that faster completion times would lead users to prefer interacting out-of-turn. Therefore, we conducting a comparative study in which participants performed two different sets of similar tasks in a voicemail, menu-based IVRs: one with the ability to interact out-of-turn and one without (hereafter called *baseline*). We evaluated differences in the above factors using common protocols and instruments from HCI such as questionnaires and the SUS (System Usability Scale) [3].

System Configuration

We administered a questionnaire to 151 undergraduate students which revealed that respondents were extremely familiar with and frequently use ATS, including voicemail. Therefore, for purposes of familiarity, we decided to conduct our study within the domain of voicemail. Furthermore, in order to insulate our study against the nuances of a particular commercial voicemail system as well as make our results more generalizable, rather than employing a commercial system, we designed a menu-based, voicemail IVRs [7] specifically for use in our study. While there are several commercial voicemail systems available, each with minor variations, idiosyncrasies, and facilities (e.g., some have guest mailboxes while others have delivery options), they all have a common set of core functions (send a message, listen to messages, change password, and so on). In order to include in our system a representative cross-section of the landscape of the features and functions available in voicemail systems, we based our system on a survey we conducted of approximately 10 commercial voicemail systems, including Siemens *PhoneMail*, Nortel Networks *MerdianMail*, and Verizon *VoiceMail*. We culled common functions and terms from these systems to design a representative voicemail decision tree (see Fig. 5). We implemented the voicemail system using VoiceXML and hosted two instances of it — out-of-turn and baseline — in the *BeVocal Café*, a free web-based service which hosts VoiceXML applications on the Internet, interprets them using *Nuance* automated speech recognition (ASR) technology, and provides toll-free access to them.

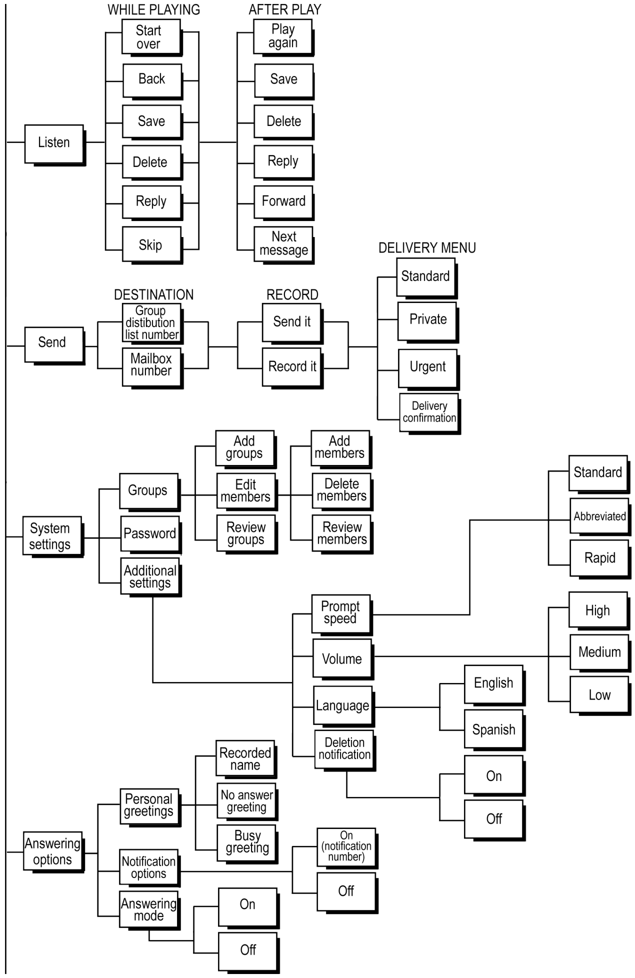

Figure 5. Menu structure for the voicemail IVRs created for our study.

Experimental Design, Participants, and Procedures

Our study employed a *mixed-factorial design with order counter-balanced*. We only analyzed data from participants who had never interacted with systems similar to those using out-of-turn interaction (including systems which use pure natural language), determined through an exit questionnaire. While 46 undergraduate students (23 females and 23 males) participated in the experiment, we disqualified six (3 female, 3 male): five for this reason and another for not completing questionnaires according to the instructions. Due to the *Nuance* ASR engine used in *BeVocal*, which requires no training, only naive speakers of English, 18 years of age or older, were permitted to participate. Each participant was paid $10 and each session took approximately one hour. The mean participant age was 20.6 and 77.5% of participants were psychology majors.

Half (10 females and 10 males) of the 40 participants were exposed to the *baseline* interface first followed by the *out-of-turn* interface, and vice versa for the other half. Participants were assigned to an order of interfaces based on the order in which each signed-up to participate. This design permitted us to evaluate performance (on the first interface used across all subjects) and preference (between the two interface conditions across all subjects) in a single experiment. Moreover, it did not preclude us from evaluating performance on the first *and* second interface used across all subjects should we not observe any learning from the first to second interface condition, therefore increasing the power of the statistical tests involved.

Since interacting out-of-turn is optional in an interface, the out-of-turn interface subsumes the baseline. Therefore to foster an effective comparison, we instructed participants to interact out-of-turn from the main menu only when performing the tasks within the out-of-turn interface condition. Recall that our goal was to measure the effect of out-of-turn interaction and not to evaluate whether or not participants employed it when presented with a task which might benefit from its use. At the beginning of each interface condition, we gave participants instructions on how to interact. Before using the out-of-turn interface, participants were instructed that they could ignore the current prompt and say up to three words in a single out-of-turn utterance rather than responding directly to a prompt. Participants were also told that they could say 'main menu' at any point in the interaction using either interface to return to the home state.

Training and Tasks

Immediately before performing any tasks with a particular interface, each participant listened to an approximately half-minute recording of a sample dialog between a user and a menu-based, IVRs demonstrating how a user could find the weather conditions in Boston. We used two instances of this dialog: the one to which the participants listened prior to using the baseline interface involved only solicited responses, while that to which they listened prior to using the out-of-turn interface involved out-of-turn utterances. The prerecorded dialogs were not created from the system used in this study; they were only intended to illustrate the interaction technique, and not to help the user learn the structure of the menus they would navigate to perform the actual tasks.

Each participant performed 12 tasks in total during the session. While they performed 6 in each interface condition, the first two tasks attempted in each were practice and common across each condition. Therefore, our study involved 10 distinct tasks, of which 8 were experimental.

Practice Tasks

Participants performed the following two tasks for practice.

- (2-step) You have reason to believe that someone has learned the number that protects your phone information. You need to change your password to make sure no one has access to your protected phone information.
- (4-step) You wish to add a member, John James, to your phone-calling group. The phone-calling group allows you to leave voicemail messages simultaneously in the mailboxes of those on your calling list.

We categorized tasks based on the optimal number of steps necessary to successfully complete each using the baseline interface. We define a step as a transition from one menu to a sub-menu. We also define successful task completion to mean simply reaching the sub-menu relevant to the task at hand. For instance, the first practice task requires 2-steps: one from the home state to the 'system settings' menu and one from there to the password facility. The second practice task requires at least 4 steps to complete: from the main menu to 'system system settings' to 'groups' to 'edit members' and finally to the 'add members' menu. We annotate each task below with its value for this metric, which provides a measure of complexity. These practice tasks were presented to each participant in both interfaces conditions in the order shown above.

Experimental Tasks

The experimental tasks used in our study were:

- (2-step) Recently you have been receiving messages from unknown people and you need to prevent your mailbox from accepting messages. Turn your answer mode to off.
- (2-step) This system is currently configured so that you will not be notified when you have a new message. Since you want to be notified, turn the notification on.
- (3-step) You wish to change the message people hear when you are busy with another call. Change your busy greeting to: 'Please call back later.'
- (3-step) You wish to add a new phone-calling group which will allow you to leave voicemail messages simultaneously in the mailboxes of those in the group. Add a group named 'Group 2.'
- (3-step) The speed at which the system plays the prompts is currently set to standard, but you want the prompts to be played at a rapid rate. Change the prompt speed to rapid.
- (3-step) You want to check your voicemail messages while you are driving on the highway. The road noise is substantial so you must change the volume of your messaging system to high.
- (3-step) The language in which the prompts are played is currently set to the default of English. You would like to change the system language to Spanish.
- (3-step) The voicemail system currently asks you to confirm the deletion of a message. You find this feature annoying and want to turn it off. Turn the deletion notification off.

In order to eliminate task-based learning, each participant performed the four experimental tasks in each condition in a random order. Specifically, one 2-step task was selected randomly from the set of two 2-step tasks, and three 3-step tasks were randomly selected from the set of six 3-step tasks. Therefore, the number of 2-step and 3-step tasks was balanced in each condition. However, all 2-step tasks preceded all 3-step tasks. We feel that this matches the pattern of a novice user learning a system: they typically start with simple tasks and gradually move up to more difficult tasks.

Participants were not given the menu structure of the system (Fig. 5) or any other assistance at any time during the experiment. During the experiment as well as in all documents, the two interface conditions were referred to as *red* and *blue*, not baseline (or in-turn) and out-of-turn. Participants were instructed that they had 5-minutes to complete each task.

Participants used a standard cordless telephone to access the system. We recorded the audio from each participant session using a digital voice recorder and used a stopwatch to measure task completion time from recorded audio files. We started timing the task at the start of the menu main prompt. We stopped timing each task either when the participant arrived at the menu relevant to the task at hand or hung up the phone.

After completing each experimental task, each participant rated the interface for that task on four factors (easy/difficult, simple/complex, usable/unusable[2], not-frustrating/frustrating) using 6-point (bipolar) semantic differential scales. Immediately after performing the final task with each interface, participants completed an interface questionnaire followed by the SUS. At the end of the experiment participants completed an interface comparison questionnaire followed by an exit questionnaire.

RESULTS

Successful Task Completion Rate

Our experimental design involved 320 (40 participants × 8) experimental tasks attempted. Of these 320 trials, 305 (95.31%) were completed successfully. Of the 15 remaining trials, eight were not completed within the five-minute time limit. On the other seven trials, participants hung up the phone before completing the task (though they each thought that they had completed the task at the time that they hung up). Of the 15 unsuccessful trials, 11 involved the 'notification on' task, 2 involved the 'deletion notification' task, and one each came from the 'change busy greeting' and 'change answer mode off' tasks. Both unsuccessful attempts at the 'deletion notification' task involved prematurely hanging up within the baseline interface condition. Each unsuccessful attempt at the 'change busy greeting' and 'change answer mode off' involved an early hangup, using the baseline interface for the former and the out-of-turn interface for the latter. Overall, four of our eight experimental tasks were completed successfully by all participants. Ten participants did not complete one task, one participant did not complete two, and only one did not complete three. A deeper analysis of the 15 unsuccessful trials indicates that participants were not likely to complete tasks more often in one interface than another. We shall have more to say below about the 'notification on' task – that which had the lowest task success rate.

The task with the longest successful mean task completion time ('notification on') took 78s. Those participants who exceeded the 240s limit may not have fully understood the capabilities of the system or were confused by the menus. Those participants who hung up before completing the task may not have read the task carefully enough or may have confused one task for another. These unsuccessful trials are not related to a specific interface as seven were not completed with the baseline interface and the remaining eight trials not completed with the out-of-turn interface. We only analyzed time and semantic differential data from participants who completed the specified task successfully.

		Task-completion times				
		baseline		out-of-turn		
Task	n	μ	σ^2	μ	σ^2	Δ
Answer mode off	39	69.85	58.58	23.05	32.86	46.80*
Busy greeting	39	74.68	29.15	33.80	53.34	40.88*
Add a group	40	36.40	11.83	14.35	6.46	22.05*
Prompt speed	40	44.65	15.61	10.25	2.59	34.40*
Raise volume	40	38.95	9.10	14.70	12.88	24.25*
Change language	40	55.05	30.32	13.25	8.55	41.80*
Del. notification	38	57.56	22.23	17.30	18.42	40.26*
Notification on	29	78.00	44.68	72.46	69.81	5.54

Table 1. Successful task completion time means (μ) and standard deviations (σ^2) in seconds by task. Key: $*$ denotes significantly different at $p < 0.01$ level.

Task	Preferred OOT	$\chi^2(1)$	p
Answer mode off	77.5%	12.10	=0.001
Busy greeting	70.0%	6.40	=0.011
Add a group	87.5%	22.50	<0.001
Prompt speed	85.0%	19.60	<0.001
Raise volume	90.0%	25.60	<0.001
Change language	90.0%	25.60	<0.001
Del. notification	75.0%	10.00	=0.002
Notification on	67.5%	4.97	=0.027

Table 2. Interface preference by task.

Task Completion Time

To determine whether the order in which our experiment exposed participants to the two interfaces had an effect on successful task completion time, we conducted a 2×2 (order × interface type) ANOVA on mean task completion times for each of the eight experimental tasks. We found no significant interaction effect (p >0.05) between conditions on task completion time on any task. This is important because it meant that we could analyze completion times without regard to the order in which participants used the interfaces, thus, substantially increasing the power of the statistical tests. Therefore, we performed a 2×2 ANOVA[3] on mean successful task completion times. We found a significant (p <0.01) main effect of interface type for seven of the eight tasks (see Table 1). *This is noteworthy because it means that all tasks, except for the 'notification on' task, were completed significantly faster, on average, while using the out-of-turn interface than the baseline interface*. While participants completed the 'notification on' task faster on average using the out-of-turn interface, we may have observed an insignificant difference in mean times because only 29 participants successfully completed the task, thus reducing statistical power. We shall have more to say about this task below.

Preference

Eighty-five percent of all participants significantly favored the out-of-turn interface over the baseline ($\chi^2(1)$=19.60, p <0.001). Of the 20 participants who interacted with the out-of-turn interface first, 95% significantly preferred it over the baseline ($\chi^2(1)$=16.20, p <0.001). Of the 20 participants who used the baseline first, 75% significantly favored the out-of-turn interface over it ($\chi^2(1)$=5.00, p=0.025). Moreo-

[2] While the SUS captures *overall* system usability, here we measure usability on *each* of the experimental tasks using a semantic differential scale.

[3] Since the distribution of task completion times was significantly different from a normal distribution, we also used the Mann-Whitney U test for non-parametric statistical significance. However, since the patterns of significant differences (p <0.05) of the eight tasks were the same for the ANOVA and the Mann-Whitney U, we present the results of the ANOVA.

ver, results indicate that for all eight experimental tasks, the out-of-turn interface was significantly preferred to the baseline (see Table 2). The differences in the percentages could be attributed to the idea that an initial exposure to a quicker interface makes the slower interface seem much slower, than if users begin with the slower interface first.

Usability: SUS

When the baseline interface was used first, the mean SUS score for it and out-of-turn interface was 65.38 and 85.38, respectively. When the out-of-turn interface was used first, the mean SUS score for it and the baseline interface was 72.13 and 66.75, respectively. We conducted a 2×2 (order × interface type) ANOVA on mean SUS scores and found no significant interaction effect of the order of interface presentation ($F(1, 76)$=2.955, p=0.090). Therefore, we examined all SUS scores without consideration of the order in which our experiment exposed participants to each interface and found no significant main effect of order on SUS scores ($F(1, 76)$=2.638, p=0.108). However, we did find a significant main effect of interface type on SUS scores ($F(1, 76)$=9.456, p=0.003) indicating that participants found the out-of-turn interface significantly more usable than the baseline (whether they used the out-of-turn interface first or second).

Ease, Simplicity, Usability, and Frustration

Using the Mann-Whitney U test for non-parametric statistical significance, participants rated the 'answer mode off' task significantly simpler (U=111.00, p=0.026) and significantly more usable (U=117.50, p=0.041) with the out-of-turn interface than the baseline. Similarly, participants rated the 'deletion notification' task significantly more usable (U=105.50, p=0.028) with the out-of-turn interface. Overall, participants rated the out-of-turn interface easier, simpler, more usable, and less frustrating than the baseline on 7 of the 8 tasks, with the exception of the 'notification on' task – the only task for which participants rated the baseline interface (insignificantly) simpler.

Problematic Task

The results obtained from the 'notification on' task (on all dependent variables) did not follow the result pattern from other tasks. Since 11 of the 15 unsuccessful trials involved this task, it was problematic to 11 of the 40 participants. While mean task completion times showed that participants successfully completed this task faster using the out-of-turn interface, the difference was not significant as in all the other tasks, and this task also had the highest mean task completion time (μ=78s). Similarly, participants significantly preferred to use the out-of-turn interface over the baseline on this task, but the preference percentage (67.5%) was the lowest percentage of all the tasks. Lastly, on each of the four semantic differential rating scales, mean ratings showed that participants found the baseline interface easier, simpler, more usable, and less frustrating for this task; the opposite was found on all of the other (seven) tasks.

There are several possible explanations for this result. The unfamiliar nature of the 'notification on' task may have confused some participants. Only two participants in the entire sample had ever changed the message notification setting in their own voicemail account. Another explanation may be the duplicate use of the term 'notification' in the system. In the baseline interface, those who navigated to the deletion notification setting first may have assumed that this option was the same as the notification option. However, saying 'notification' out-of-turn would only eliminate two options from the main menu ('listen' and 'send'), and participants would therefore still have to make a selection between the two remaining main menu choices: 'system settings' and 'answering options.' This decision is also present in the baseline interface, which may explain why the mean completion times in the out-of-turn interface (μ=72.46s) and baseline (μ=78.0s) are relatively similar. In the other seven tasks, the initial out-of-turn utterance from each participant resulted in a shortcut to task completion. For instance, during the 'change language' task, participants most often said either 'change language' or 'language' – either utterance brings users directly to the language setting facility, obviating the need to make a subsequent choice at the main menu. Another possible reason for difficulty might be that the task scenario itself may have been unclear or participants may have misread the scenario, again confusing it with the 'deletion notification' task.

In summary, the experimental results indicate that

- participants completed 7 (87.5%) of the 8 tasks significantly faster using the out-of-turn interface than the baseline,
- the order of interface exposure did not significantly affect mean task completion time,
- overall preference (85% of participants) significantly favored the out-of-turn interface and significantly more participants preferred it for each individual task,
- based on SUS data, participants found the out-of-turn interface significantly more usable than the baseline, and
- participants had difficulty successfully completing tasks when their initial out-of-turn utterance did not result in a shortcut.

Also, note that we do not report any speech recognition errors because they were too low (<1%) to be meaningful.

DISCUSSION

To the best of our knowledge, we contribute the first formal, scientific user study establishing an upper bound on the impact of out-of-turn interaction on task efficiency in menu-based, IVRs. Out-of-turn interaction is a technique which provides the user with the option to make utterances which are unsolicited from the current menu, but constrained by the choices available in subsequent sub-menus. While a hybrid of restricted and natural language, the utterances made using this technique are always out-of-turn (i.e., unsolicited). Therefore, our study also explored a new dimension (nature of user-addressable input – in-turn vs. out-of-turn) within the conceptual design space for ATS (see Fig. 4).

Out-of-turn interaction is not a substitute for a well-designed IVRs menu, since it preserves the modes of navigation originally modeled in the menu design accessed through in-turn means, as evidenced by the difficulties involved with the 'notification on' task. Such tasks will require further study. Moreover, to be effective, out-of-turn interaction requires users to have a basic understanding of the general nomenclature of the underlying domain (banking, travel) [17]. However, the results of our study have provided substantial evidence that out-of-turn interaction significantly reduces task completion time, improves usability, and is preferred over fixed menus, thus confirming our original expectations. These results can be used to make an informed decision on whether or not to support out-of-turn interaction within an existing IVRs tree. Moreover, armed with the current results, we can study the effect of relaxed assumptions and constraints in future studies.

The ubiquity of IVRs in a variety of service-oriented domains (banking, travel) provide fertile ground for the application of out-of-turn interaction and our results, especially to reduce agent costs in call routing. For these reasons we feel that our study is worthwhile and particularly timely.

ACKNOWLEDGMENTS

This work was supported in part by a seed grant from the University of Dayton Research Council and the Ohio Board of Regents. We thank Robert Capra for implementing the voice interfaces used, Jack Deaton at *BeVocal* for system support, and James Tramontana at UD for helping with figures.

REFERENCES

1. J. F. Allen, C. I. Guinn, and E. Horvitz. Mixed-Initiative Interaction. *IEEE Intelligent Systems*, Vol. 14(5):pp. 14–23, 1999.

2. B. Arons. Hyerspeech: Navigation in Speech-Only Hypermedia. In *Proceedings of the Third Annual ACM conference on Hypertext (HT)*, pp. 133–146, 1991.

3. J. Brooke. SUS: A Quick and Dirty Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *Usability Evaluation in Industry*, pp. 189–194. Taylor and Francis, London, 1996.

4. A. L. Gorin, G. Riccardi, and J.H. Wright. How May I Help You? *Speech Communication*, Vol. 23:pp. 113–127, 1997.

5. V. Lakshmipathy, C. Schmandt, and N. Marmasse. TalkBack: A Conversational Answering Machine. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pp. 41–50, 2003.

6. A. Mane, S. Boyce, D. Karis, and N. Yankelovich. Designing the User Interface for Speech Recognition Applications. *ACM SIGCHI Bulletin*, Vol. 28(4):pp. 29–34, 1996.

7. M. A. Marics and G. Engelbeck. Designing Voice Menu Applications for Telephones. In M. Helander, T.K. Landauer, and P.V. Prabhu, editors, *Handbook of Human-Computer Interaction*, pp. 1085–1102. Elsevier, Amsterdam, 1997.

8. P. Resnick and R. A. Virzi. Skip and Scan: Cleaning Up Telephone Interface. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 419–426, 1992.

9. P. Resnick and R. A. Virzi. Relief from the Audio Interface Blues: Expanding the Spectrum of Menu, List, and Form Styles. *ACM Transactions on Computer-Human Interaction*, Vol. 2(2):pp. 145–176, 1995.

10. M. Ringel and J. Hirschberg. Automated Message Prioritization: Making Voicemail Retrieval More Efficient. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 592–593, 2002. Interactive Poster.

11. B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, MA, Third edition, 1998.

12. S. Srinivasan and E. Brown. Is Speech Recognition Becoming Mainstream? *IEEE Computer*, Vol. 35(4):pp. 38–41, 2002.

13. B. Suhm, J. Bers, D. McCarthy, B. Freeman, D. Getty, K. Godfrey, and P. Peterson. A Comparative Study of Speech in the Call Center: Natural Language Call Routing vs. Touch-tone Menus. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 275–282, 2002.

14. S. Tomko, T.K. Harris, A. Toth, J. Sanders, A. Rudnicky, and R. Rosenfeld. Towards Efficient Human Machine Speech Communication: The Speech Graffiti Project. *ACM Transactions on Speech and Language Processing*, Vol. 2(1):pp. 1–27, 2005.

15. S. Whittaker, R. Davis, J. Hirschberg, and U. Muller. Jotmail: a voicemail interface that enables you to see what was said. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 89–96, 2000.

16. S. Whittaker, J. Hirschberg, B. Amento, L. Stark, M. Bacchiani, P. Isenhour, L. Stead, G. Zamchick, and A. Rosenberg. SCANMail: A Voicemail Interface that Makes Speech Browsable, Readable and Searchable. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 275–282, 2002.

17. N. Yankelovich. How Do Users Know What To Say? *ACM Interactions*, Vol. 3(6):pp. 32–43, 1996.

18. M. Yin and S. Zhai. The Benefits of Augmenting Telephone Voice Menu Navigation with Visual Browsing and Search. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 319–327, 2006.

19. W. Zadrozny, M. Budzikowski, J. Chai, N. Kambhatla, S. Levesque, and N. Nicolov. Natural Language Dialogue for Personalized Interaction. *Communications of the ACM*, Vol. 43(8):pp. 116–120, 2000.