Working Memory, Search, And Signal Detection: Implications For Interactive Voice Response System Menu Design

2006

Patrick Commarford
University of Central Florida

WORKING MEMORY, SEARCH, AND SIGNAL DETECTION: IMPLICATIONS FOR INTERACTIVE VOICE RESPONSE SYSTEM MENU DESIGN

by

PATRICK M. COMMARFORD
B.A. University of Kentucky, 1994

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Psychology
in the College of Arts and Sciences
at the University of Central Florida
Orlando, Florida

Spring Term
2006

Major Professor: Janan Al-Awar Smither

ABSTRACT

Many researchers and speech user interface practitioners assert that interactive voice response (IVR) menus must be relatively short due to constraints of the human memory system. These individuals commonly cite Miller's (1956) paper to support their claims. The current paper argues that these authors commonly misuse the information provided in Miller's paper and that hypotheses drawn from modern theories of working memory (e.g., Baddeley and Hitch, 1974) would lead to the opposite conclusion – that reducing menu length by creating a greater number of menus and a deeper structure will actually be more demanding on users' working memories and will lead to poorer performance and poorer user satisfaction. The primary purpose of this series of experiments was to gain a greater understanding of the role of working memory in speech-enabled IVR use. The experiments also sought to determine whether theories of visual search and signal detection theory (SDT) could be used to predict auditory search behavior. Results of this experiment indicate that creating a deeper structure with shorter menus is detrimental to performance and satisfaction and more demanding of working memory resource. Further the experiment provides support for arguments developed from Macgregor, Lee, and Lam's dual criterion decision model and is a first step toward applying SDT to the IVR domain.

I dedicate this dissertation to my wife, Christie, who provided a tremendous amount of support and encouragement, while *rarely* complaining of the demands this endeavor made of my time. I appreciate all of her help in making this accomplishment possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Interactive Voice Response Menu Length – Current Design Guidelines

Automated phone-based user interfacing systems known as interactive voice response (IVR) systems have become a ubiquitous solution for the purpose of reducing customer support costs. These IVR systems allow users to accomplish many of their goals without the help of a human representative, allowing the organization to reduce support staff. Many individuals, however, hold negative attitudes toward this technology, often characterizing IVRs as being difficult to use. Therefore, it is extremely important that IVR design is optimized to allow users to accomplish their goals effectively and efficiently, leading to user satisfaction and acceptance.

Researchers and speech user interface practitioners commonly assert (e.g., Balentine & Morgan, 2001; Cohen, Giangola, & Balogh, 2004; Gardner-Bonneau, 1992; Schumacher, Hardzinski, & Schwartz, 1995) that interactive voice response (IVR) menus must be relatively short due to constraints of the human memory system. These individuals often cite Miller's (1956) paper to support their claims that humans simply cannot remember more than 7 +/- 2 items and, for this reason, argue that speech user interface designers should find ways to limit the number of items in any particular menu. For example, Cohen et al. write:

> …Miller (1956) described a pattern of human short-term memory in which people can store seven, plus or minus two, items. Often designers use this as a guideline for the

> number of items to put into lists, menus, and so on. However, listening to sentences over the phone while trying to extract and remember information from these sentences is much more taxing than the tasks Miller used in the lab. The caller's task is more akin to the listening task in which subjects are asked to listen to a series of sentences and remember the last word of each sentence (Daneman and Carpenter, 1980). In experiments using this completely auditory approach in which sentence comprehension is also taking place, people can remember only about three items on average.
>
> Other research on human memory has shown that people naturally cluster items in threes and that recall is best when information is divided into groups of three or four items (Broadbent 1975; Wickelgren, 1964). Taken together, the research results suggest that the caller's memory load should be kept quite small. A reasonable guideline is to limit menus to three or four items.

In agreement, Gould, Boies, Levy, Richards, and Schoonard (1987) and the Voice Messaging User Interface Forum (1990) advocate no more than four options per menu. Marics and Engelbeck (1997) also state that menus should be limited to four or fewer items, but advise that items such as Help and Exit should be excluded from this count. Others (e.g., Devauchelle, 1991, as cited in Schartz and Hardzinski, 1993) advise limiting options per menu to no more than three. Schumaker, Hardzinski, and Schartz (1995) make an exception for certain types of information. They mention that, for example, a movie list would be preferably played as a single list and they follow with: "Thus, for command-like options, limit the number of items on a menu to four. For object-like options (e.g., movies, names), in which each item can be rapidly dismissed, the list can be as long as nine items" (p. 257). Gardner-Bonneau (1999) also acknowledges that a movie or similar type of list could be an exception to this rule. Although Schumaker et al. (1995) and Gardner-Bonneau (1999) recognize lists as exceptions; there is widespread agreement in the speech user interface community that IVR menus should be kept short to avoid overtaxing users' memories, especially for command-like menu items.

It's important to note that the IVR design question under consideration is relevant only to situations in which there is a group of more than a few commands that fit reasonably well together. The features and functions of an IVR system should be defined before the design phase and should be based on analyses of the range of goals of the potential users. Since, at design time, the features and functions are fixed, it is the designer's role to determine how to best allow users to access these options. A primary concern is that the designer places the items into logical groups to meet users' expectations and avoid confusion. If the items fall nicely into groups of four or fewer, it is widely accepted that they should be organized in this manner. The single long versus multiple short menu design decision becomes relevant when more than a few items are highly-related or are all relevant at a particular point in the user interface flow.

To illustrate this design point, consider a scenario in which a patron is using a speech-enabled IVR to order a sausage pizza. The IVR might prompt the user to, "Select meats, vegetables, or fruits." The patron would select, "Meats." Now, at this point, assuming the pizza company offers six or more meat toppings, the designer has the option to provide all meat toppings here (longer, broader structure) or to provide another menu that further classifies the meat toppings (for example, “Select beef, pork, or fish”), with each higher level menu providing access to a few items (shorter, deeper structure). The shorter, deeper structure would not have any menus with more than a few items, but would also require additional user interaction and navigation to access and select “Sausage.”

IVR Input Method

There are two primary methods of input for IVRs: speech and dual tone multiple frequency (DTMF; also referred to as “touch tone” or “keypad” input). These two types of interfaces impose different requirements on users’ cognitive resources. DTMF systems require that users key a number or symbol on the telephone dial pad to make a selection and speech-enabled user interfaces allow users to interact with the system through speech. Since DTMF systems pair a number or symbol with each selection, this input method imposes additional demands on the user. For example, if recall of the options is indeed a goal (as discussed immediately above, many authors make this assumption), then DTMF systems require users to store twice the amount of information (the keypad selection and the command) as a speech-enabled system (command only). Further, whereas speech user interfaces often offer predictable input selections, users of a DTMF system must almost always wait to hear which keypad number or symbol is associated with the desired selection (one exception to this is the pairing of the zero key with “help” or “to speak to a representative,” which is highly predictable). Further, stimulus-response modality compatibility (Wickens & Hollands, 2000) suggests that user performance will be different when IVR user responses are provided through voice than by key press. For example, Brainard, Irby, Fitts, and Alluisi (1962) found that individuals can respond faster vocally than manually to an auditory cue. Also, Wickens, Sandry, and Vidulich (1983) and Wickens, Vidulich, and Sandry-Garza (1984) found evidence leading to the conclusion that the combination of auditory output and voice input are best suited for tasks that require use of verbal working memory (e.g., an IVR task). This paper focuses on cognitive resources and strategies associated with speech-enabled IVR use.

Working Memory, Search, and Signal Detection Theory

There are three areas of psychological theory and research that are particularly relevant to IVR use. These topics are working memory and capacity, auditory and visual search, and signal detection theory (SDT). The following sections provide information about our current understanding of these theories and processes in general. Later sections describe specifically how these concepts relate to IVR use.

Working Memory Theory & Measurement

Although the authors mentioned previously commonly cite Miller's (1956) paper as the definitive source for how the human memory systems functions, our current knowledge of the human working memory has evolved from research pre-dating Miller's paper and continuing to the present date. The following is a review of this research.

Where it all began. Miller's (1956) paper indeed indicated that individuals can typically recall somewhere from five to nine items and that this number can be increased by compounding items into groups (incidentally, contrary to popular belief, this is not a concept that Miller *discovered* through experimentation, rather one that was considered common knowledge among cognitive psychologists prior to his writing the famous paper). Miller cited experiments conducted by Hayes (1952) and Pollack (1953) that strongly indicated that the amount of information available for immediate recall increases substantially as the amount of information per item increases. The number of items that can be recalled changes to a relatively small degree as compared to the change in amount of information recalled. As the amount of information per item increases, the number of items available for immediate recall is attenuated, but not linearly (in fact, the count remains remarkably constant). For example, Hayes found that the memory span for binary items is about nine, but the memory span for monosyllabic words is approximately five (five monosyllabic words contain much more information than nine binary numbers). Miller used the term "chunks" to describe units of information, but he recognized that the definition or boundaries of a "chunk" are not clear-cut. He argued that people build larger and larger chunks through a process called recoding. We take smaller pieces of information, group them, and give the group a name, creating a single chunk containing more information. He proposed that this is the process by which individuals learn a new language. Miller's paper had great influence in that it *pointed out,* among other interesting observations, that memory span is, on average, between five and nine items for most "chunks" of information.

Waugh and Norman's model. Waugh and Norman (1965) theorized a two-component, memory system, consisting of a "primary memory" and a "secondary memory." Primary memory was analogous to a short-term store and secondary memory to a long-term store. Regarding primary memory capacity, they make the statement, "It is a well-established fact that the longest series of unrelated digits, letters, or words that a person can recall verbatim after one presentation seldom exceeds 10 items" (p. 89). Waugh and Norman wanted to investigate *why* unrehearsed items are forgotten. Specifically, they were interested in determining whether the memory trace for the item gets overwritten by new information (interference theory) or simply decays with time. They presented participants with a set of digits at a rate of either one per second or four per second and instructed participants to think only of the most-recently presented digit during the inter-digit interval. At the second presentation of any digit, the list ended and participants were to attempt to recall the digit that had immediately followed the initial presentation of this "probe digit." Since participants didn't know in advance which digit would be the probe digit and because the experimenters instructed them to think only of the most recent digit, the digit-pair information was unrehearsed. Waugh and Norman found that, under these conditions, recall was approximately equal regardless of the presentation rate and that, as the number of intervening items increased, recall decreased. Since reducing time did not increase memory, but reducing intervening items did, they reasoned that the unrehearsed information is forgotten through interference and that memory trace decay is not a valid theory. Waugh and Norman theorized that, if rehearsed, information held in primary memory can transfer to secondary memory. They also theorized that information can reside in each memory component simultaneously. Like Miller (1956) before them, they acknowledged that an "item" has no clear definition or boundaries, but acknowledged a capacity of fewer than 10 items.

Atkinson and Shiffrin's box model. Atkinson and Shiffrin (1968; 1971) describe what has come to be known as the "standard model" or the "box model" of memory store. They theorized that we have a sensory store that holds information for .5 – 1 second, a short-term store that can hold information for several seconds to several minutes, and a long-term store (LTS) that can hold information, potentially, forever. Atkinson and Shiffrin theorized that the duration of the information in the STS is dependent on whether interfering information enters the STS and on whether the individual uses strategies, such as rehearsal, to retain the information. Consistent with their predecessors, Atkinson and Shiffrin argued that the short-term store has a limited capacity for information and that as new information is received, old information is removed – individuals cannot simply continue to add information to this limited store. Information held in the short-term store will be lost unless it is rehearsed either through maintenance rehearsal (repeating it over and over again) or elaborative rehearsal (relating the information to something meaningful) – in which case the information can be transferred to the long term store. The view that there is a temporary memory system that can hold a limited amount of information, then, is a view that was shared by Miller (1956), Waugh and Norman (1965), and the box model advocates.

Craik & Lockhart's levels of processing. Craik and Lockhart (1972) argued that, rather than consisting of a countable number of discrete stores, memory is made up of a potentially infinite number of levels. They proposed that these levels range from very shallow to very deep, and that information is encoded and processed within these levels. They agreed that the human memory system has a limited capacity, but argued that this is determined by depth to which information is processed. If information is not processed at a deep enough level, they claimed, it would be lost; and it is not possible to process a large amount of information if not combined meaningfully. Though Craik & Lockhart disagreed with their predecessors that there is a discrete short-term memory store, they agreed that individuals are not able to hold a large amount of information without meaningfully processing the information and that meaningful processing cannot occur with a large amount of data instantaneously. The original Craik and Lockhart (1972) article received some criticism (e.g., Eysenck, 1978; Nelson, 1977) with a primary complaint being the circular nature of the theory. Specifically, it is easy to claim that anything that is well-remembered has been processed at a deep level, with ability to recall being the unit of measurement for the item. In a 1978 article, Craik and Lockhart agree with some of the criticisms, stating that, "…the notion of depth of processing by itself is insufficient to give an adequate characterization of memory processes" (pp. 174).

Baddeley and Hitch's working memory system. Baddeley and Hitch (1974) proposed a working memory system that is far more active and complex than that described by Atkinson and Shiffrin (1968; 1971). The introduction of their model marked a shift from theories of a short-term memory store to theories of a working memory system. Their original model proposed that working memory is composed of three main systems: a) the central executive, b) the phonological loop, and c) the visuospatial sketchpad. The central executive is the key component of this working memory system and is said to control and monitor one's attention according to current needs. The phonological loop (also known as the articulatory rehearsal loop) stores and rehearses speech-based information and registers visual information in the phonological store. The rehearsal process is referred to as "subvocalization" or "inner speech," a process by which the loop repeats the phonetic information every few seconds to avoid losing it. The visuospatial sketchpad sets up and manipulates visual images. According to Baddeley and Hitch's model, the phonological loop would be the system responsible for one's ability to process, store, and recall items presented via auditory means.

Baddeley and Hitch's (1974) description of the phonological loop is able to account for many findings related to individuals' ability to retain information, such as the acoustic similarity, irrelevant speech, word-length, and articulatory suppression effects. The acoustic similarity effect is an effect by which recall is poorer for a list of words that sound alike than for dissimilar sounding words (Conrad & Hull, 1964). Baddeley (1999) explains that this is because the code is phonological and therefore the fewer the number of distinguished features, the more difficult to recall. The irrelevant speech effect is an effect by which irrelevant speech acts to reduce the amount of information that can be recalled, even when the stimulus is presented visually (Colle & Welsh, 1976); whereas loud non-speech noises do not exhibit this effect on recall (Salame & Baddeley, 1987). Likewise, repeating an irrelevant word reduces recall (Murray, 1968) – an effect known as articulatory suppression. Each of these activities (listening to and repeating irrelevant speech) presumably interrupts the articulatory rehearsal process. The word length effect describes a phenomenon such that individuals are able to recall fewer words as the individual list items increase in length (Baddeley, 1999). The explanation is that, as word length increases, it takes longer and longer to rehearse the words; therefore making it more difficult for the individual to maintain the information through the rehearsal process. Baddeley and Hitch's original working memory model was revolutionary in that it stipulated that the working memory system is responsible for both storage and processing of information.

In more recent years Baddeley (Baddeley, 2000; 2001), has added the concept of an episodic buffer, which he postulates is able to combine information from the central executive and the long-term memory. He also proposes direct communication between the slave systems (the phonological loop and the visuospatial sketchpad) and long-term memory. Baddeley used evidence from many years of varied auditory, visual-spatial, and multimodal tests of memory span (e.g., Baddeley & Lieberman, 1980; Baddeley, Thompson, & Buchanan, 1975; Chincotta, Underwood, Abd Ghani, , Papadopoulou, & Wresinski, 1999; Colle & Welsh, 1976; Conrad & Hull, 1964; Hulme, Roodenrys, Brown, & Mercer, 1995; Jones 1993; Logie, 1986; Quinn & McConnell, 1996;) as well as neurological evidence (Prabhakaran, Narayanan, Zhao, & Gabrieli, 2000) to revise the model to its current form.

Just and Carpenter's capacity theory of working memory. Just and Carpenter's (1992) capacity theory also views working memory as an active system that is responsible for more than simple storage. Just and Carpenter conceptualize "activation" as a single commodity that allows storage, retrieval, and computing. They conceptualize capacity as the amount of activation available. In this model, information can become “activated” by perceiving and encoding the information, computing it, or retrieving it from long-term storage. At levels above a certain threshold of activation, a particular bit of information becomes part of working memory, making it available for operation via cognitive processes. Information is forgotten when activation must be reallocated to process information or perform a task. Activation allocation is prioritized for current operations; therefore storage of non-essential information can be forgotten when this reallocation occurs. This model differs from Baddeley’s (2000; 2001) model in that it does not propose discrete systems and components, instead favoring an activation pool that can be used for such processes as information storage and computation.

Working memory capacity. Working memory capacity (WMC) is the measurement of the amount of storage and processing capacity available to an individual. Traditional short term memory capacity tests required users to perceive a series of digits or words and then repeat these items back once the stimulus is no longer present (Reisberg, 1997). As they are able to complete each set, the experimenter presents larger and larger item sets until the participant starts making mistakes. This original method of measuring capacity is often referred to as a short-term memory span task. It is these tests that provided evidence that, in general, users can hold five to nine items in working memory (e.g., Hayes, 1952; Keller, Cowan, & Saults, 1995; Pollack, 1953; Smyth, Pearson, & Pendleton, 1988; Watkins, 1977).

More modern theories (e.g., Baddeley and Hitch, 1974; discussed above) view working memory as an active system, responsible for more than simple storage. The working memory system is now assumed responsible for directing attention resource, processing and operating on information. Many activities humans engage in require both storage and processing capabilities. Arithmetic problems, for example, require this more complex set of working memory processes. For example, to calculate the sum of two three-digit integers, individuals must store information, operate on it, and derive new information for temporary storage and operation. Traditional tests of short-term memory capacity (memory span tests) measure storage capacity only and do not account for these types of working memory processes. Therefore, as theory developed, researchers began developing measures of working memory capacity that take these processes into account.

Daneman and Carpenter (1980) introduced the first complex span task, the "reading span" task. They asked participants to read a set of 2 – 6 unrelated sentences and then attempt to recall the final word in each sentence. This task combines a traditional memory span task with additional processing demands. Engle, Nations, & Cantor (1990) developed another complex span task they referred to as an "operation span" task. For this task, the experimenter presented a series of math equations followed by a word or digit. The participants verbally indicated whether each equation was correct or incorrect and then attempted to remember the set of words or digits that followed each equation. For example, participants might see, "(3x3) – 4 = 4; Dolphin" followed by "(8/2) + 1 = 5; Bottle." In this case, the participant would respond "No" to the first equation and "Yes" to the second. Once all equation/word pairs have been presented, the participant attempts to recall all the words. Many others (e.g., Cantor & Engle, 1993; Carpenter, Just, & Shell, 1990; Engle, Cantor, & Carullo, 1992; Just & Carpenter, 1992; Kane et al., 2004) have used a variation of a complex span task. Unsworth, Schrock, Heitz, and Engle (2003) developed an automated version of an operation span task, which is administered via a computer terminal and is automatically scored. Barret, Tugade, and Engle (2004) report that, "Confirmatory factor analytic evidence (that controls for measurement error) suggests that people perform consistently across a host of different span tasks that require different types of computations to be made" (p. 556). These types of complex span tasks are more valid than the traditional span tasks since they are closer in nature to the tasks for which we rely on our working memory systems on a daily basis (Reisberg, 1997) and have been demonstrated to be correlated with reading comprehension scores on standardized tests (Baddeley, Logie, & Nimmo-Smith, 1983; Reisberg, 1997; Shah and Miyake, 1996; Turner & Engle, 1989) as well as measures of fluid intelligence (Engle, Tuholski, Laughlin, & Conway, 1999) and attentional control (Kane, Bleckley, Conway, & Engle, 2001).

Working memory conclusions. Although there is debate regarding the specific underlying mechanisms involved, there is widespread agreement that there is an upper limit to the amount of information an individual can hold in "short-term" or "working" memory. Depending on context, five to nine items is a good rule of thumb for estimating the upper boundaries of simple memory span. However, modern theories of working memory (e.g., Baddeley, 2001; Just & Carpenter, 1992) support a system that goes beyond simple short-term information storage, suggesting that working memory capacity be measured by the ability to store and process information. When attempting to simultaneously operate on information, a smaller amount can be stored.

Auditory and Visual Search

Visual search is an activity by which an individual scans an area visually in attempt to locate one or more target items. In the typical case, the participant searches for a single target item. The participant may be engaging in an identity matching, equivalence matching, or class-inclusion matching search. The search can also be classified as serial or parallel in nature and can be self-terminating, exhaustive, or redundant as described below.

Identity matching occurs when the individual knows precisely what the target is and can compare each item to the known target held in working memory (Paap & Cooke, 1997). Equivalence matching is a search technique by which the user knows the target, but does not know its precise characteristics. For example, the user is searching for a target that will allow him to make a change to some entity, but he doesn't know if he's looking for precisely, "change" "edit" "manage" "modify" or a similar label. Since the target is not known, equivalence matching takes longer than identity matching. As Paap and Cooke describe, "…the user…must engage in a slower semantic analysis of the relationship between the target and each option" (p. 534). Class-inclusion searches are searches conducted to find a particular class to which the target item belongs. The purpose of a class-inclusion search is to help the individual navigate one step closer to locating the target item (for example, selecting “animal” when the target is “eagle”).

A distinction can be made between parallel and serial visual searches. A parallel search is one in which the individual inspects all items in the display simultaneously; when searching serially, on the other hand, the individual inspects each item one-by-one. For serial searches, the time to detect the target will increase as the number of intervening items (distracters) increases. If all items are displayed simultaneously, the individual can search the display in parallel or serially. However, when data is displayed sequentially, it is not persistent; therefore, the search must be serial. The individual can conduct a serial search that is either self-terminating or exhaustive. That is, individuals can stop searching when they believe, with high enough confidence, that they have encountered the target item (self-terminating search) or they can search each item until the entire display has been exhausted before making a selection (exhaustive search). Moreover, when there is a single target, users may engage in a redundant search. In this case, the user views each item in the display and then returns to one or more items to re-evaluate them against the target they are holding in working memory before making a decision.

Auditory search is an activity by which an individual aurally scans a group of items for one or more targets. These search tasks may require individuals to scan for target syllables (Davis, 1967), target letters (McGuinness, 1983) target sounds (Dalton & Lavie, 2004; McGuinness, 1983) or target words (Frederiksen, 1969). As with visual search, individuals can engage in identity matching, equivalence matching, and/or class-inclusion matching. Auditory search behavior can also be self-terminating, exhaustive, or redundant as described above. Since a stream of auditory information must display serially and the information is not persistent, auditory searches of any given stream cannot be conducted in parallel (the presentation restricts this). However, it is possible to present information to an individual through multiple streams, by using more than one audio source. In this case, the individual may attempt to search the streams in parallel; however, if each auditory stream is more than one item (target or distracter) deep, a true parallel search is not possible.

Humans are capable of performing effectively on some rather complex auditory search tasks. For example, participants have shown the ability to perform well in polychotic listening tasks in which they must search multiple simultaneous auditory streams for a target word (Lee, 2001). In Lee's experiment, participants monitored a set of two to six speakers for a target word and source location with the target being presented within 10-word lists. Participants had to identify and respond to a target item being presented simultaneous with at least one other item and within a total of at least 20 items. Lee found that, without practice, participants could not only identify an auditory target stimulus, but also identify from which audio stream the target was generated 80% of the time when the number of sources was held to two. Even when presented with six channels (60 total list items, presented 6 at a time) users were able to correctly identify the item and source location 26% of the time.

As previously discussed, when searching serially for a single target item (e.g., in a menu), users can conduct self-terminating, exhaustive, or redundant searches. MacGregor, Lee, and

Lam's (1986) criterion-based decision model offers an explanation as to why individuals choose each search strategy under a given context, assuming the potential target items are presented sequentially; therefore the criterion-based decision model describes user behavior that is dictated by comparisons between each item and its degree of relationship with the target item. This theory applies specifically to class-inclusion searches (the type of search that occurs at a higher lever menu), but could be extended to equivalence-matching activities as well. This criterion-based decision model will be discussed in more detail in a later section which discusses the relationship between search strategies and IVR menu use.

Signal Detection Theory

People rely on a great deal of signals to help us effectively and safely manage our daily lives. The dryer buzzes, the tea kettle whistles, the traffic light changes from green to yellow to red, the phone rings, and countless other signals are presented to us throughout a given day. Often, due to low signal strength or high noise levels (internal or environmental), these signals are more difficult to detect than those described above. For example, it may often be difficult to detect the ring of one's own cell phone from others' on a crowded subway. Signal detection theory (SDT) describes the processes by which individuals determine whether a target signal is present or absent.

In a typical signal detection task, users are introduced to a target item and then must detect whether the item is present or absent over a series of trials. This requires that users maintain characteristics of the target in working memory as they search. A "hit" is recorded for each trial in which the participant correctly states that the target signal is present. Each time the participant fails to acknowledge the target signal when it is present, the experimenter records a "miss." If participants indicate a signal is present, when, in fact, it is not, a "false alarm" is recorded. Finally, for trials in which the participant appropriately indicates that no signal was present, a "correct rejection" is tallied. Figure 1 illustrates this set of signal detection outcomes.

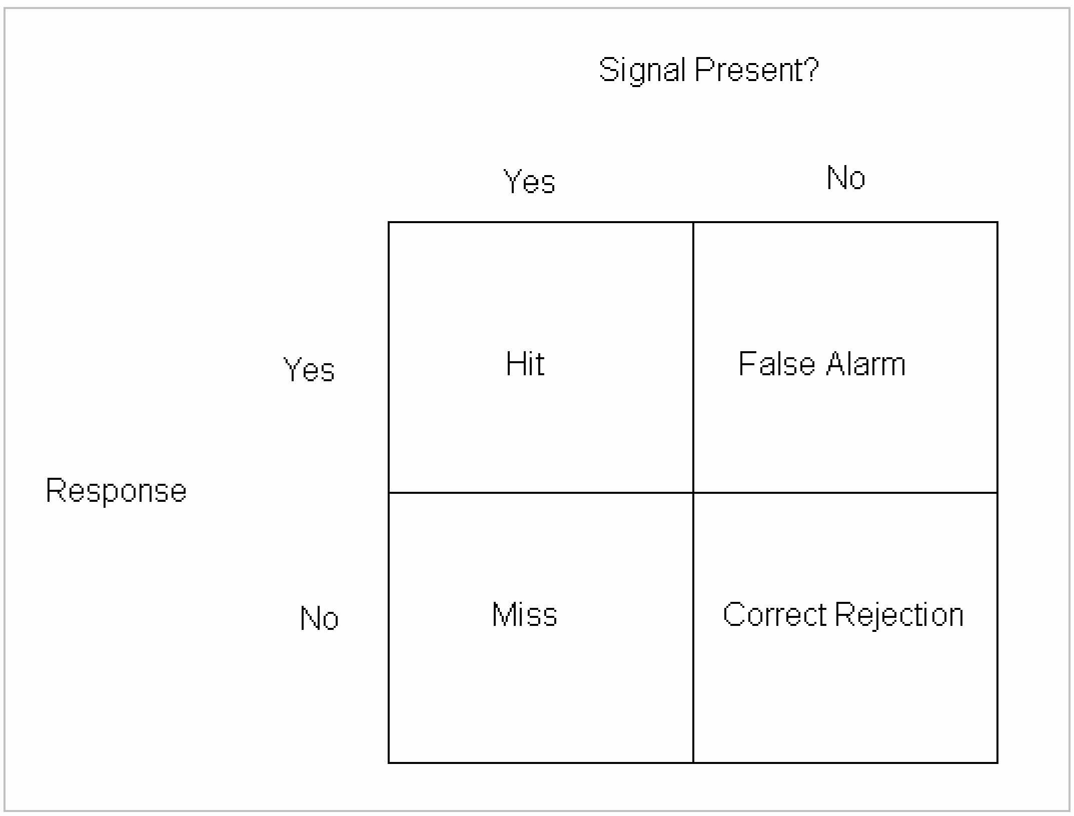

Figure 1. Signal detection theory outcomes

In attempt to determine whether a signal is present or absent, participants aggregate sensory evidence and then make a decision as to whether the evidence is sufficient to determine that the signal is present (Green & Swets, 1966). According to Green and Swets, external stimuli (signal and environmental variations) as well as the individual's baseline neural firing combine to create the total amount of neural evidence on any given trial (referred to as the "evidence variable"). The individual's decision as to whether to indicate the target signal was or was not present depends on the total amount of evidence that a signal appeared and the individual's critical threshold or "response criterion." In some cases, noise alone will be great enough to warrant a positive response and, in others, the signal will not exceed the noise by a great enough level as to warrant such a response. The greater the difference between the amount of evidence on a typical signal trial and a no signal trial, the easier it is for the participant to make the right decision.

The participant's decision that the signal is or is not present relies not only on the amount of evidence, but also on the participant's response bias. A number of factors contribute to the response bias, which runs on a continuum from extremely liberal to extremely conservative. Those who employ a liberal response bias are more likely to indicate that the signal was present – leading to more hits, but also more false alarms. On the other hand, conservative responders are reluctant to indicate the presence of a signal unless the evidence is very strong – leading to many correct rejections, but also many misses. Response criterion, c (a commonly-employed bias measurement) is calculated by summing the z scores for hit and false alarm rate and dividing by – 2 (Macmillan and Creelman, 2005). This method standardizes the hit and false alarm rates using the normal distribution function.

"Beta" refers to the ratio of neural activity produced by signal and noise that occurs at the decision criterion and is influenced by signal probability and by the costs and benefits associated with each of the four outcomes (hit, miss, false alarm, and correct rejection) over a series of trials. The more likely it is that a signal will occur, the more liberally participants will set their decision criterion. For example, if participants knew that the signal would never appear (0% probability that the signal is present), they would never indicate that they believed the signal to be present. On the other hand, if they knew it would appear on the majority of trials, they would often indicate that it is present. Payoffs (consequences and rewards) associated with hits, misses, false alarms and correct rejections will also influence the point at which an individual sets beta. In an extreme example, suppose there were a task in which there were tremendous rewards for hits and no consequence for false alarms. Further, assume great consequences for misses and no rewards for correct rejections. The individual would be extremely liberal, indicating that the target signal was present at every trial. Wickens and Hollands (2000) provide the Equation 1, which illustrates how beta would be optimally adjusted for probability and for payoffs:

$$\beta_{opt} = \frac{P(N)}{P(S)} \times \frac{V(CR) + C(FA)}{V(H) + C(M)} \tag{1}$$

Where β_{opt} is optimal beta, P(N) and P(S) are the probabilities of noise and signal respectively, V is value, C is cost, and CR, FA, H, and M represent the four outcomes of a signal detection trial. Experimental evidence suggests that participants adjust beta in response to these factors, but to a less than optimal degree (Bisseret, 1981; Harris and Chaney, 1969 as cited in Wickens & Hollands, 2000;). This mismatch is more pronounced when probability is manipulated than when payoffs are manipulated (Green & Swets, 1966).

"Sensitivity" refers to the individual's ability to discriminate the target signal from environmental and internal noise. Sensitivity depends on characteristics of the signal and noise distributions and characteristics of the observer. The greater the separation between the signal and noise distributions, the more sensitive, on average the observer will be. The signal may be difficult to detect if the strength is weak or if noise is strong. The signal may also be difficult to detect if the individual's detection abilities for the specified signal are weak. Individuals may also have difficulty remembering the exact characteristics of the signal, which can lead to difficulty in detection due to an inability to discriminate signal from noise. Therefore, sensitivity increases if participants are reintroduced to the target stimulus (Wickens & Hollands, 2000). Sensitivity is measured by *d'*, which is calculated by normalizing the proportion of hits and false alarms and finding the difference (Macmillan and Creelman, 2005). Therefore, a high hit rate and a low false alarm rate will produce high d', which indicates that the observer is very sensitive.

Parasuraman, Masalonis, and Hancock (2000) describe fuzzy signal detection as a methodology for analyzing decision-making data when the state of the world and/or the response set is "fuzzy." To be fuzzy, one or both of these variables must be continuous rather than discrete. For example, weight of an item could range continuously from the lightest object in a set to the heaviest. A set of items, for example, might range in weight from 2 ounces to 400 pounds, with a tremendous number of items residing at the levels between. The participants' response could also range continuously. Rather than simply indicating that each item is or is not "heavy," the participants could assign a value from 0 to 1 to indicate how heavy the item is.

Psychological Theory and IVR Use

Reducing the Number of Menu Items

Often there are more than a few items that are relevant options at a particular point in an IVR system. Providing all of these items at such a point helps ensure that the system matches users' mental models, allowing for predictability of items and reducing confusion that could result from unfulfilled expectations. However, as described above, the traditional belief is that there is a need to limit the number of items in an IVR menu to a low number to avoid over-taxing users' working memory systems. To investigate which menu structure is more demanding of user resources and more likely to result in errors and dissatisfaction, the process a speech user interface designer should undergo in attempt to follow a guideline that restricts IVR menu items to four or fewer items should first be examined.

One method of splitting a large menu into smaller menus is to play the first three or four options of a menu, followed by a general option that provides navigation to the next set of relevant options (e.g., “more” or “next”). Schumacher, Hardzinski, and Schwartz (1995) advocate use of this method for DTMF systems. This method, however, increases the total number of options and the navigation complexity and does not receive much credit in the field – especially as a method for reducing the length of speech-enabled IVR menus. Further, Virzi and Huitema (1997) found that splitting menus by this means increases the time it takes for users to select the desired option. The primary method of splitting a long menu into sets of shorter menus is to divide the items into two or more groups and provide access to these groups via higher-level menu items (Paap & Cooke, 1997). This method reduces the number of items in each menu, but makes the menu deeper hierarchically.

Analyzing the Task and User Goals

User interfaces should be designed with the users’ goals and the necessary cognitive processes to accomplish these goals in mind. Speech user interfaces are no exception – the designer should be aware of the predominant user goals and required processes and should create dialogues and navigation that allow users to accomplish these goals as quickly and easily as possible.

User goals. Balentine and Morgan (2001) list a number of potential service provider and organization goals for an IVR system, including being fast (inexpensive), fun, easy to learn, professional, and even the ability to keep users on the line for as long as possible (when users are paying for time; note that this goal would be in direct contrast to users' goals). However, users primarily want systems that allow them to accomplish their goals quickly and easily. The large majority of users interact with a voice response system to a) efficiently complete a transaction (e.g., transfer $20 from savings to checking), b) efficiently obtain information (e.g., get the score of the Bears game), or c) efficiently obtain information and use that information to complete a transaction (e.g., find out how much money is in checking and then transfer money from savings to checking).

When recall is a priority, it is the *content* that users wish to remember, rather than the IVR command set. A user, for example, may wish to obtain and remember a co-worker's phone number or the show times for the two movies she is considering for that night. However, users do not access an IVR system with the goal of retaining menu items. Long-term retention of menu items can be beneficial for users who repeatedly access a certain IVR and short-term retention may be valuable within a single session to allow faster navigation. Although, recalling menu items may not be a user goal itself, this does not dispute the traditional thinking that recall of items is important for IVR use for information and transaction-based goals. The next sections will discuss specifically the implications of working memory, search strategies, and signal detection theory on user behavior, performance, and satisfaction associated with single long and multiple short menu IVR systems.

Working memory and IVR. Many authors feel that the ability to immediately recall all items in each menu is a key cognitive process for speech interface use. As previously discussed, these authors claim that users will be unable to work effectively with an IVR system if the system is designed in such a way to limit users' ability to recall all items in any particular menu. However, when analyzing the requirements for selecting a menu item from a list, the ability to recall all menu items does not seem essential. Instead, it seems perfectly reasonable that users should be able to discard menu items that do not match their goals as these items are presented. However, if it is necessary for users to hold all menu items in working memory, then as the number of items in an IVR menu increases, performance and user satisfaction ratings should decrease.

This paper proposes that working memory is utilized during IVR tasks in a manner such that use of a single, long menu as opposed to a set of shorter menus will not increase the demand or overtax one's working memory. Users access an IVR with a goal and then search for the target options that will help them accomplish their goal. True, it would be more difficult for a user to remember all items in a long menu than in a short menu, but there is no need to remember all items to make effective use of an IVR system. Instead users must maintain no more than a few items (generally no more than two items) in working memory before making a selection, regardless of menu length. Further, separating the items into smaller sets of menus does not reduce the total number of options; instead it just increases complexity.

In this simplest case, the user knows precisely the target option – in this case the task is akin to listening for one's name when on a waiting list at a restaurant. The hostess might call 40 names before calling the waiting patron's name, but this will, in no way, interfere with the patron's ability to recognize his own name when it is called. Further, there is clearly no need for the patron to hold all names that came before and after his in working memory. To be sure, the patron is not capable of doing so, but this is unimportant when considering whether he can act on his name when called. Similarly, when a user attempting to obtain a baseball score, for example, reaches the menu that lists the team names as options – hearing other team names will not affect the user's ability to recognize the team of interest when it is presented (or her ability to select it before it's presented) and there is certainly no need for her to hold all team names in working memory.

Examining a more complex, and perhaps more common case, suppose the user desires to obtain information or complete a transaction, but is unsure how to navigate through the menus to get to her terminal point. The user must engage in one or more class-inclusion searches before accessing the appropriate option. Referencing the baseball score example from above, when the user first calls the system, she is likely to hear a menu that offers choices such as "News," "Weather," "Stocks," "Horoscope," "Movies," "Lottery," and "Sports." This example is set up for the worst case scenario in terms of potential demands on working memory, in that the selection the user needs is buried at the end of the list, (see subsequent discussion on menu item position). This paper proposes that, in cases such as this, users select a "best of" item and hold this item in working memory. As they are presented with each additional item, they process the new item and either discard it or replace the “best of” item with this new item. They do this until they are confident that the current “best of” item will help them accomplish their current goal or until the menu ends. When either of these conditions occurs, the user makes a selection. In the example, the user might hold “News” in working memory until “Sports” is presented, at which point she would evaluate “Sports” as a better match, drop “News” and make her selection. This strategy requires that users hold up to two items in working memory at any given moment (the “best of” item and the currently-presented item). More accurately, it requires that the user hold one item (the current “best of” item) and process information about another (the menu item under evaluation). Figure 2 graphically depicts this theory of user interaction with interactive voice response systems. There may be instances in which users hold two approximately equal candidate items in working memory as they process the additional items. In these cases, when users reach the list end, they may return to re-examine the candidates before making a selection. However, in general, users will not carry two candidates, making a quick comparison as each item is presented and dropping the less attractive item.

Given that holding one or two menu items while processing another is clearly within the capabilities of normal human working memory system (Daneman & Carpenter, 1980; Hayes, 1952; Kane, et al., 2004; Pollack, 1953), this theory of IVR menu use suggests that selecting an item from a single auditory menu should not be a task that is likely to overtax users’ working memory systems. If users discard items one-by-one, rather than holding them until making a selection, then increasing the number of menu items should not decrease performance or user satisfaction.

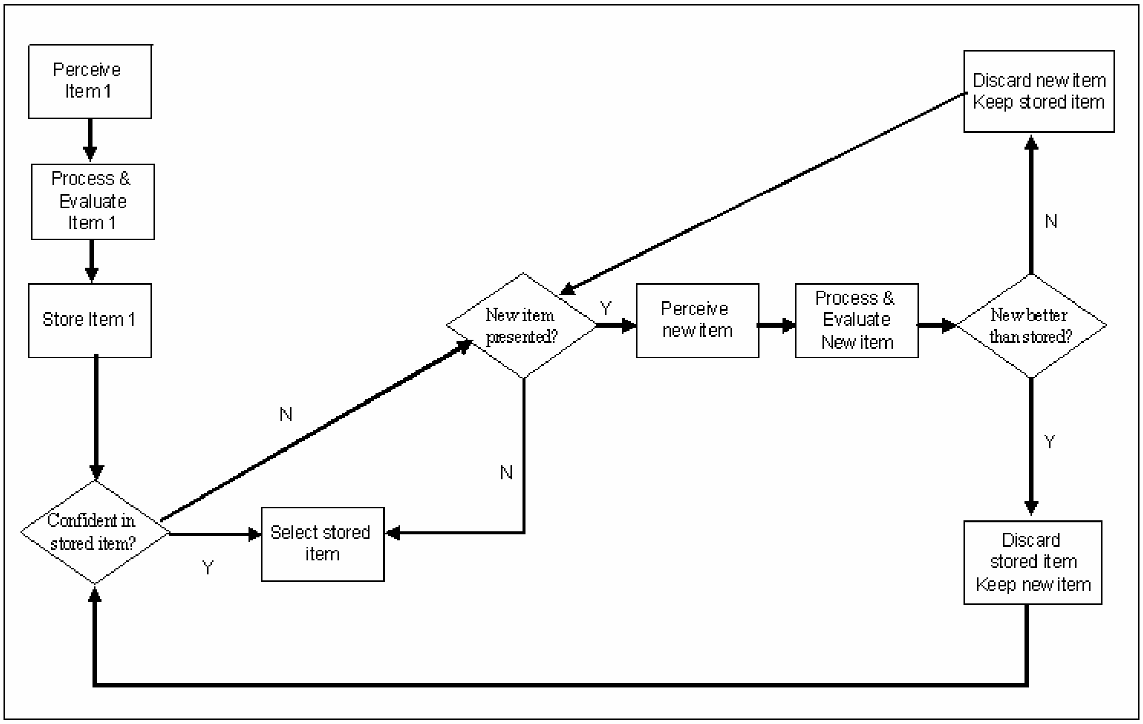

Figure 2. User flow for single item selection from single menu

This paper proposes that working memory plays a role in IVR task performance, but in such a way that increasing the number of items per menu will not overload the user's system. There are other reasons why menus should not grow to enormous lengths (matching the user's mental model, efficiency, etc.,) but, since the proposal is that users only need to hold one or two items in working memory while processing another item, increasing menu length should not cause cognitive overload. In fact, as will be argued, splitting and artificially shortening IVR menus is more likely to decrease performance.

The traditionally assumed advantage of the menu-shortening technique described by Paap and Cooke (1997) is that, by limiting the number of menu items in each menu, the system will not overload the user's working memory. However, dividing menus into shorter sets by the preferred method also has potential disadvantages. For example, the split scheme may be inconsistent with users' mental models and expectations, leading to confusion and mistakes. As Paap and Cooke also explain, "In a well-designed menu system the category names are highly distinctive from one another and there will be little overlap between the distributions…" (pp. 542). It is clear that if a set of similar items are purposefully distributed across multiple menus with the intent of avoiding a single long menu, cognitive overlap will be likely.

As previously alluded to, another potential problem with splitting menus in this manner is that it will likely actually increase the demand on working memory by creating a need that is not present with longer menu sets. The total number of menu items to be evaluated does not shrink; the deep menu just makes it more difficult to access each of these items. Paap and Cooke (1997) describe that deeper visual menus require users to recall or discover how to get from where they are to where they want to go, which leads to users getting lost and traversing inefficient pathways. If an IVR menu is split and deepened, users may need to traverse the system to hear all options for a particular point. Now users have a more demanding task when keeping their "best of" item or items (they have to retain the best of item while traversing the IVR). Colle and Welsh (1976) and Jones (1993) found that immediate serial verbal recall is impaired by presentation of auditory material, even if participants attempt to ignore the material. It is also well known that the repetition of an irrelevant speech degrades recall performance (Baddeley, 2000). Splitting a single menu into subgroups with a higher level menu requires users to a) hear additional speech, b) produce additional utterances, and c) meaningfully process additional material. This degrades the user's ability to hold the "best of" item or items and evaluate newly perceived items. Further, when users have exhausted all options, they must traverse back to the menu that will allow them to make their decided upon selection. This new demand requires that users remember where the to-be-selected option appears. So, the user needs now to store the "best of" item as well as the higher level menu item and navigation path while processing information about new items.

Based on these analyses, it is likely that a system designed with a set of deeper menus, as opposed to a single long menu, will generally be more demanding on users' working memory systems and will demand a greater amount of user interaction. Therefore, the hypothesis derived from these analyses is that a deeper structure will lead to a greater number of errors, increased time on task, and decreased user satisfaction ratings. Further, any task that tests users' working memory systems will be more taxing for those with lower working memory capacity. Those with low WMC may or may not exhibit decreased performance and satisfaction as compared to their high WMC counterparts using a long menu system, but there will likely be a noticeable difference in the performance and satisfaction of low and high WMC users when attempting to accomplish tasks with a deep structure. In other words, there will be an interaction such that a deep, short menu structure will be detrimental to all, but more so to those with low WMC.

Search, Signal Detection, and IVR. When users know the specific menu item for which they are searching, the task is one of simple identity-matching (though, depending on the system it may be preceded by a series of class-inclusion matches). Individuals should have no problem detecting the target word in a single IVR menu if they know precisely what the target is (a simple identity matching task). When the target is known precisely, IVR tasks are relatively easy because they require that users only scan for a single meaningful word or phrase. Further, these words or phrases are typically acoustically distinct from the other items to allow for reliable recognition by the speech system (Balentine and Morgan, 2001). The auditory search literature contains many examples of individuals completing auditory tasks that are much more difficult from a cognitive and sensory perspective than simple detection of a target word within a stream of other words from a single source. As will be discussed, complexity is introduced to the search task when users must first determine *where* the target item resides, engaging in an initial class-inclusion search.

For identity matching tasks, a broad menu with several items will be a much better design than multiple short menus with a higher level menu. There are several reasons for this. With a broad menu, once the user reaches the point at which the target selection becomes valid, there is no navigation and no guesswork. All options are available from a single IVR location and the user simply needs to speak the appropriate selection. A multiple short menu system, on the other hand, requires first a class inclusion search, and then a search for the target within the class. So, the short menu system adds the need for an additional search and this added search activity alone is more difficult and time consuming than the original identity matching task (Giroux & Belleau, 1985). Identity matching is the sole requirement when using a single menu. Indeed Snowberry, Parkinson, and Sisson (1983) found that participants engaging in visual search tasks with 64-items distributed across a single page (64^1), in 2^6, 4^3, and 8^2 arrangements, performed better (in terms of speed and accuracy) as the menu became broader (for the single page, this relationship breaks down if the items are dispersed randomly as opposed to categorically on the page). This information supports the working memory literature in a prediction that broader IVR menus will produce better performance.

Also, many speech-enabled IVR systems have "barge-in" enabled, which means that once users reach the menu within which the target item resides, they can interrupt the prompt and speak ahead, making the selection before the IVR presents it. This is a variation of the self-terminating search in which the user actually terminates the search before the target item is presented. When users spend time navigating to other parts of the menu structure, they cannot simply speak the target item, since the item will not appear in the grammar at that point – even though the presented options are related. If the user does not or cannot speak ahead, the broad system will still require less navigation, will require the user to listen to fewer total options, and will require fewer user and system speech turns – each of these factors will work to decrease the amount of time on task and the number of errors. The increased need for user interaction (increased number of speech turns) can lengthen time on task even with perfect system recognition accuracy. However the reality is that speech-enabled IVRs do not exhibit perfect recognition and that users commit speech errors. As the number of required system-user communication turns increases, so will the total number of errors.

Often, however, when working with an IVR, the user does not know precisely what the target menu item is and must engage in an equivalence matching task. The user knows the general semantic qualities, but is unsure of the precise target word or phrase (for example, the user may be searching for something like “parts” “accessories” “add-ons” “extras” or a synonym). The nature of many IVR tasks, then, is one of equivalence matching. This activity is similar to the case in which the user has been presented with a signal (e.g., a tone) before the trials of a signal detection task begin and is now unable to remember the precise characteristics of the target signal. In each case, the user is attempting to discriminate the signal from noise while sensitivity is decreased (because the user is unsure of the precise characteristics of the target). In a tone task, the participant cannot remember the acoustic qualities; in the IVR task, the user is aware of the semantic qualities of the word or phrase, but is unaware of the precise phonetic characteristics. This reduces the user’s sensitivity and leads to a task that is more difficult than simple auditory search and detection (identity matching). For these equivalence-matching IVR searches, users’ behaviors can be dictated to some degree by the menu design. MacGregor, Lee, and Lam’s (1986) criterion-based decision model and signal detection theory are each able to offer insights as to the search strategies and types of errors to which each menu structure will likely lead.

MacGregor, Lee, and Lam’s (1986) criterion decision model states that individuals create low and high criterion levels. Any option that falls beneath the low criterion is immediately rejected. Any option that falls above the low criterion, but below the high, is considered a candidate and is maintained in working memory. Finally, any item that falls above the high criterion is immediately selected, resulting in a self-terminating search. If, after all items are presented, only a single candidate (item falling between the low and high criterion) exists, the individual will choose that item, resulting in an exhaustive search. When an individual must choose a single candidate and has encountered multiple items that fall between the low and high criterions, the individual will often revisit the candidates before making a selection (a redundant search). Paap and Cooke (1997) suggest that alternatively, the individual might simply select the best of the candidates, without revisiting them (a second route to an exhaustive search).

MacGregor et al. (1986) used the dual criterion model to make several predictions about the type of menu search users will engage in and the types of errors that will occur given short and long menus. The model suggests that as the number of options grows, the number of self-terminating searches will grow. There are two reasons for this. First, as the number of options grows a greater number of and proportion of the items appear before the final item. Therefore, a greater number of items are available to exceed the high criterion marker posed by the criterion model (resulting in search termination). For example, in a two-item list, only 50% (one item) of the items display before the list terminates with the final item. This makes a self-terminating search less likely than with a list containing, for example, 1,000 items, in which any of the 999 items that precede the final item can exceed the high criterion and cause the user to terminate the search. Secondly, the greater the number of items in the list, the greater the cost associated with reviewing all list items before making a selection and the greater the reward associated with a correct early selection. This same argument would be predicted by signal detection theory – as the costs associated with a miss and the rewards associated with hit increase, the user should switch to a more liberal strategy. MacGregor et al. (1986) supported this prediction with an experiment in which participants were presented with items visually and sequentially. The proportion of self-terminating searches increased significantly and linearly as the number of options rose, with pages containing 2, 4, 8, and 16 options. Pierce, Parkinson, & Sisson, (1992) also found that increasing the number of menu options increases the number of self-terminating searches.

The predictions made by the dual criterion and signal detection theories and supported by the work of MacGregor et al. (1986) and by Pierce, Parkinson, & Sisson (1992), suggest that longer lists lead to more self-terminating searches. However, the design point of interest and of practical concern is not how many list items to provide, but how to structure these list items. MacGregor et al. compared single menus with 2, 4, 8, and 16 items and Pierce et al. compared single menus with 2, 4, and 8 items, but neither group compared, for example, a single menu with 16 items to two pages, each containing 8 items. Converting from a single long menu to a deeper menu structure with short menus does not reduce the number of total options; rather it spreads them out. In other words, the models (dual criterion and signal detection) predict more self-terminating searches with a 15-item menu than with a 5-item menu; however, this does not suggest that displacing 15 items over three menus with a higher-level menu will reduce the number of self-terminating searches (since the number of items does not decrease—it actually increases if the higher-level menu items are tallied). Increasing navigation complexity for a set number of items should, in fact, lead to a greater number of self-terminating searches (riskier behavior), since the costs associated with examining the remaining options increases with navigation complexity. This is supported by signal detection theory and can be extrapolated from the dual-criterion model as well.

MacGregor et al. (1986) also predicted that, as the number of alternatives increases, the number of exhaustive searches will decrease and the number of redundant searches will increase. Their claim was that, as the number of options increases, users' lower bound criteria will also decrease; therefore they will accept more potential candidates, which will lead to a need for re-review (to resolve "ties"). It seems that a more reasonable argument is that, with a greater number of options, more will fall between the lower- and upper-bound criteria (there's really no reason to assume users would reduce their lower-bound criteria). The results of their experiment supported their prediction as the number of options increases from 2 – 8; however, a reversal of this effect occurred when the number of options increased from 8 – 16. They proposed that this latter effect is due to the effort required to re-review a long list. In support, Pierce, Parkinson, & Sisson, (1992) found an increase in the number of redundant searches and a decrease in exhaustive searches as the number of options increased. Their explanation is that when users complete the scan of a long list of items, they are often no longer able to remember the details of earlier options and feel compelled to return to and review these items.

It seems strange that increasing the number of options would lead to an increase in both self-terminating and redundant searches, as these two search techniques are at opposite ends of a continuum with the middle ground (exhaustive searches) decreasing. This push toward one side or the other can possibly be explained by users' varying estimations of the payoffs (rewards and consequences) associated with the four outcomes of SDT. While some users might perceive the time costs associated with examining many additional items before making a selection as more problematic than the costs associated with an incorrect selection, a second group of users take the opposite view. These differing views lead to a fundamental split in users' response criterion and push each user group away from the middle ground (the exhaustive search). The more liberal users' beliefs push them to make more early (self-terminating) selections in order to avoid the time waste associated with reviewing the remaining items; whereas the more conservative users' beliefs guide them to review all items. Once these conservative users have reviewed all items, they feel compelled to re-review certain items to ensure that their extra efforts lead to the correct selection.

At first thought, this might suggest that a single long IVR menu would lead to a greater number of redundant searches than the short menu structure. However, again, remember that the number of low-level alternatives does not change and the number of total items (low and high level) is actually smaller when using a single long menu. Therefore, based on the reasoning that users engage in redundant searches due to a loss of the ability to remember precise characteristics of one or more items, users should be at least as likely to wish to review items when using a deep structure as the broad structure. In fact, the addition of the need to navigate and to engage in and listen to speech will actually make it more difficult for users to maintain the necessary information in working memory to make a good decision – leading to a greater need for re-review (redundant searches) when using the short, deep menu structure system. However, users may actually resist redundant searches using a deep system, since the review process requires navigation, which can be demanding of time and cognitive resources. This is supported by MacGregor et al.'s, 1986 findings (the number of redundant searches increased as the list grew to 8 items and then decreased when jumping to 16 items). In other words, users may be more likely to make decisions they are less confident about to avoid the costs associated with the review process (this is a trade-off that SDT would predict). Since re-reviewing items is easier with a single, long menu, users would be more willing to review when using the single, long menu than the short, deep structure. However, if the decided upon item is clear, but not in the final menu accessed, navigation back to specific short menu that contains the desired selection will be necessary. If the desired item is not in the current menu, users will have to navigate back to the menu containing the option or choose a less appropriate option from the current menu. If a reasonable candidate resides in the final menu accessed, users might select this candidate, rather than engage in the difficult review process. In sum, the design of a deep structure makes review more important for accuracy, but the additional temporal and resource demands imposed by such a design may make redundant searches more difficult. It is difficult to make predictions as to the effect of a deep structure on redundant searches since there are competing factors (after exhausting a set of menus, there is a cost associated with review; however, once a candidate is decided upon, navigation back to the item is often required to make the selection). However, this discussion should make clear that deep structure users will, in general, be more willing to take a chance on a currently in-grammar command (lower their high criterion), to avoid the difficulties associated with navigation.

There is a strong relationship between signal detection theory and search type. Whereas MacGregor et al. (1986) and Pierce and associates (Pierce, Parkinson, & Sisson 1992; Pierce, Sisson, & Parkinson 1992) viewed the menu as a single trial item that did or did not contain the target, the following discussion applies to item-by-item analysis of user responses and assumes that each menu or menu set contains a single, correct target item and several incorrect items. As a series of auditory signals are presented, the user will hit or miss the target when presented and will either correctly reject each distracter or commit one or more false alarms. When users hit the target on initial presentation, they engage in a self-terminating search (unless the target is presented in last position). If users miss, but store the target and make the selection when the list ends, they engage in an exhaustive search. If they miss, listen to the rest of the options, then review one or more items before making a selection, they have engaged in a redundant search.

Signal Detection Theory is able to offer additional insight into users' behaviors with broad versus deep menu structures. In general, probabilities are unlikely to play a large role in response criterion for IVR search, but payoffs are presumed to have a direct influence. There are clear differences in the payoffs associated with hits, misses, false alarms, and correct rejections for each design; however, the probability of the target item appearing in the IVR is the same, regardless of menu type (though different on an individual menu basis) and participants are unlikely to assume different probabilities of signal occurrence. Further, participants shift their decision criterion more in response to a change in payoffs than to a change in probability of occurrence (Bisseret, 1981; Harris and Chaney, 1969 as cited in Wickens & Hollands, 2000). Payoffs are likely instrumental in shaping the decisions that users make as to when to a) select a target before listening to all options (self-terminating search), b) listen to all options and then make a selection (exhaustive search), or c) listen to all options and then re-examine one or more candidates before making a selection (redundant search).

When a user is searching an IVR menu, in general, there will be one and only one item in each list that moves users in the most efficient direction toward their goal. This item is the target and the other items are distracters. The user will either select each item or ignore it. There is no option for an item to be a degree of the target and no option for the user to indicate so – in other words, IVR items cannot be placed on a scale and must be selected or rejected. Therefore, fuzzy signal detection, though valuable in many other contexts, is not applicable to IVR selection tasks.

In the SDT paradigm, hits will provide greater reward for those using a deeper, short-menu system. This is because the deeper structure requires more navigation, more system speech turns, and more user speech turns to access all items, resulting in a greater amount of time and a greater number of errors. When users correctly select the target before navigating to and listening to all items, they save themselves time and prevent errors – leading to more efficient completion of goals. In turn, misses are of much greater consequence to short menu structures. When users fail to select an item at time of presentation, they are penalized by the need to continue navigating through the system and then back to the menu that contains the target item (at the completion of the final short menu, the item will not be in the speech system's grammar).

False alarms are slightly more problematic for users of long menu systems. In each case, when users make an incorrect selection, they are penalized by being routed to an area of the user interface that will not help them accomplish their goals. There will be some time associated with the user's realization that the current path is not the desired path, and this amount of time is variable, but should be considered equivalent for each menu structure. However, when users return to the menu from which they committed the false alarm, typically by using the universal "go back" command (Cohen et al., 2004; IBM, 2001), they will be returned to the first item in the menu from which they made the selection. With a short menu structure, users will have to re-listen to a smaller number of items on average (this is limited by the total number of items in the menu from which the user made a selection). Correct rejections are slightly more beneficial for users of a long menu structure following the same logic. Because hits are more rewarding and false alarms less costly for deeper structures, and because correct rejections are more rewarding and misses less costly for broader structures, those using a broad structure will select a more conservative response bias than those using a deep structure IVR.

The analysis above describes that misses are much more costly to users of a deep, short structure than to long single menu system users. False alarms are only slightly more problematic when using a single long menu than a group of short menus. Since misses are so costly for those using a short menu system (more costly than false alarms are for those using a long menu system), the costs associated with errors are greater than with a long menu system. For example, suppose a user misses the target word when it appears as an early list item (perhaps the third position in a long menu or the third position of the first of a set of three short menus). Using a single long menu, the user is punished by the need to review the remaining six options, and then, while holding the correct option as the best of or as the premier candidate, the user simply speaks the selection. The only consequence is the need to listen to the remaining items before making the selection. However, in a short menu system, if the user is not confident of the target option when presented, he must then engage in much navigation to be presented with the other candidate options. Upon deciding that he is now confident of his choice, he must navigate back to the menu containing the desired option before making the selection. Likewise, assuming the user wants to re-review two or more options before making a selection, his task will be much easier when working with a single menu (especially if the deeper structure splits the to be re-examined items over more than one previously-visited menu). Since the error conditions lead to harsher consequences with a short and deep structure, SDT joins modern theories of working memory and search theories in a prediction that a shorter, deeper structure will lead to poorer performance and lower user satisfaction.

Response latency. Parasuraman and Davies (1976) describe that the further the observation point of the signal from the response criterion, the quicker individuals will respond. Therefore, if the criterion is placed at a point such that a greater proportion of the responses are correct than incorrect, then correct responses should be delivered more quickly. In agreement, the literature concerning response latencies associated with signal detection tasks generally indicates faster correct than incorrect responses (Bonnel & Noizet, 1979; Pike, Dalgleish, and Wright, 1977). For an IVR task, only latencies associated with overt responses (hits and false alarms) can be measured. It is expected that mean hit response latency will be smaller than mean false alarm response latency.

Psychological Theory and IVR Use - Conclusions

The psychological theory discussed above leads to a number of hypotheses. Our current knowledge of the human working memory system suggests that users will be more effective and more satisfied when using an IVR system that has a single long menu of appropriately grouped items than a system that artificially separates these items into multiple short menus in a deeper structure. This is because a) the split scheme may be inconsistent with users' mental models, b) the added need to navigate the user interface to access all options will decrease users' ability to maintain information in working memory because it requires users to speak, listen to speech, and process information, and c) when users decide to select an item they will often need to recall where the item resides as well as recalling the command itself, because the command will not be in the other menu grammars. Further, the performance and satisfaction degradation experienced by users of the deeper structure should be exacerbated for those with low working memory capacity. This is because any system that tests the limits of working memory will exceed the limits more often and to a greater degree for low-capacity individuals.

An understanding of MacGregor et al.'s (1986) dual criterion model and of signal detection theory leads to a number of additional hypotheses about the behaviors in which users will engage. Analysis of these theories also reinforces the hypotheses derived from working memory research (stated above). First, it seems clear that, when engaging in an identity matching IVR task, a single long menu will be superior. This is because breaking the items into several short menus adds an additional, more difficult class inclusion search to the task – greatly increasing task complexity. It can also be reasoned that the deeper structure with a set of short menus will lead to more self-terminating searches. This is because the consequences associated with a miss or with setting an extremely high upper-bound criterion are greater when the task of examining the remaining items is difficult. The additional navigation associated with the multiple menu system makes this more difficult. Those users who wish to review every item before making a selection (those who believe false alarms to lead to greater consequences) will be more likely to benefit from a redundant search when using short menus because the navigation and additional user and speech turns will interfere with the maintenance of item characteristics in working memory. Further, redundant searches will be often be necessary when using a short, deep system since the desired item is likely not in grammar at the final list terminal point. Although redundant searches will be needed more often with the short, deep structures, users may still be less likely to engage in these searches – once again because the review process is more difficult than with a long system. In these situations, as compared to a long menu system, short menu users will have to make a decision that will result in more errors (exhaustive search) or spend more time to make the correct decision (redundant search).

The magnitude of the costs and rewards associated with hits, misses, false alarms, and correct rejections for each design leads to a hypothesis that users of long menu systems will adopt a more conservative style than deep, short menu users. False alarms will generally be more costly for users of long menu systems; however, the consequence of a miss will be much greater for short menu users. Since a miss has a greater cost for deep menu structures than a false alarm does for a broad system, performance will, in general be better when using the long menu system. Finally, better performance and greater user satisfaction is expected with a single long menu than with a set of short menus because of the increased interaction necessary. Increasing speech turns will always lead to an increase in time on task and will generally lead to a greater number of speech and speech recognition errors (exacerbating the time-on-task increase).

The Role of Target Location

Another variable that can be important is the position of the target item within a menu. When users are unable to predict (or remember from past experience) the appropriate speech input for the desired menu item, it takes longer to listen through a list of 10 items than to only listen to the first 2 items in a list of 10. Note that the common belief that long menus are too taxing of users' working memory either does not take target location into account or assumes that users must attempt to recall all items regardless of the position of the desired item.

Although Paap and Cooke (1997) advise the general use of broad rather than deep menu visual displays, they cite three reasons why one might want to use a deeper structure. The first reason is to avoid crowding (sometimes referred to as visual clutter), a concept that is not relevant to auditory displays. The second is to insulate rarely used or illegal menu items. This technique is practiced in IVR design regardless of the structure of the menus that provide the main items. For example, many speech-enabled IVR systems offer users the ability to switch to a DTMF system, but the option is generally not displayed until a series of speech errors occur. The final reason is to funnel, which means that the designer provides shorter pathways to certain items. For late position items in an IVR, it's possible that a set of shorter menus (creating a deeper structure) will allow faster navigation to a target item. For example, an item might be 15th in a single menu or 3rd in a menu that requires an additional step to access (this requires a new, higher-level menu that will contain a set of menu items as well, but it is clear that there would still typically be a savings for accessing this item associated with splitting the menu). For early position items, it's unlikely that a newer, deeper scheme would be beneficial, because it forces an additional turn to access an item that would otherwise appear early. Though shorter pathways to late position items can be beneficial, they also require more user interaction, which can lead to an increase in time and errors (Paap & Cooke, 1997; Snowberry, Parkinson, & Sisson 1983), so the trade-off requires careful consideration.

Depending on the number of items in the higher-level menu, the degree to which the menu structure is understandable and matches users' expectations, the degree to which the items in the new higher-level menu are recognizable, and the level of system recognition accuracy, a deeper structure could save or cost variable amounts of time for each menu item, given the item's position in a competing single, long menu. For these reasons, any investigation of the effect of menu structure on IVR usability should control for target item location. When considering results, designers and researchers should also keep in mind that good design places commonly chosen items near the front of the list and rarely chosen items at the end – therefore it's often the case that substantially fewer users will be affected by costs or savings associated with late position items than cost or savings associated with early position items.

Individual Differences

Speech-based interactive voice response systems are designed for general use – meaning that no special skills, abilities, or qualifications should be necessary to interact with them. Some systems, of course, provide content that is only meaningful to certain subsets of individuals, but use of the system itself should never require special skills. Certain disabilities, of course, will limit individuals' ability to interact with a speech-enabled IVR system. For example, users with hearing or speech deficits may have extreme difficulty or be completely unable to use such a system. Many speech-enabled IVR systems are designed to allow users to choose a touch-tone mode at the outset and some simply revert to touch-tone mode when the user and system encounter repeated communication problems. Users with strongly accented speech may also experience problems using speech-based systems since the voice recognition technology is only capable of recognizing speech within a certain range. With the aforementioned caveats, speech-based IVRs are designed for general use and should be accessible and easy to use by the great majority of people. Since these systems require no special skills and no learning or experience, they are often referred to as "walk-up-and-use" systems.

IVR systems will be used by a large and varying group of people and designers must take this into account. There are many individual difference variables that could potentially affect performance on IVR tasks. For example the physical and sensory limitations mentioned earlier, familiarity and comfort with technology, fluency of language, and various cognitive abilities could all play a role in one's ability to perform with a given IVR system.

Those with physical, perceptual, and sensory disabilities and those who fall on the extreme lower end of many individual difference variables (e.g., general intelligence) will be completely unable to use an IVR system. However, these systems are designed to accommodate the great majority of users that do not have one of the aforementioned disabilities or speech characteristics. Speech-enabled phone user interface designers should expect that, throughout a given day, week, or year, users covering nearly the entire range on individual differences such as intelligence, age, working memory capacity, comfort with technology, familiarity with technology, and familiarity with the IVR content will attempt to accomplish tasks. The IVR system should be robust to these individual differences.

Individual Differences in Working Memory Capacity

Working memory capacity is an individual difference variable that is directly relevant to the current research. The degree to which working memory is tested by an IVR system's structure will likely depend, in part, on individual differences in working memory capacity. This study will focus on two competing theories of working memory involvement for speech-enabled

IVR systems; therefore it would be very interesting to determine if participants with high working memory capacity are affected by menu length and structure differently than those with low working memory capacity. Participants with lower working memory capacities should be more sensitive to differences in IVR systems that increase the degree to which demands are made on users' working memories.

It's expected that all IVR systems impose some demand on working memory and that those with low working memory capacity will have more difficulty with an IVR system than those with high working memory capacity – unless the system demands so little working memory resource that even those on the low end are not adversely affected. The traditional view would suggest that this low-demand criterion can be achieved by limiting the number of items in all interface menus to fewer than five (the low end of simple memory span). My proposal suggests that increasing menu length will not increase the load on users' memory. Therefore, if those with lower working memory capacity have difficulty with an IVR, these difficulties will not be exacerbated by increased menu length. In fact, it is hypothesized that complications associated with a deeper structure will strain working memory and degrade performance and satisfaction to a greater degree for users with low working memory capacity.

In sum, the traditional view suggests that all users will be negatively affected as the number of menu items grows beyond four and to a greater extent for those low in working memory capacity. On the other hand, the current hypothesize presented in this paper is that splitting a naturally long menu into a set of smaller manus will actually increase the burden on working memory and will result in poorer performance for those with low working memory capacities.

IVR Menu Length – Research to Date

Huguenard, Lerch, Junker, Patz, and Kass (1997) investigated the effect of using a deeper, rather than flatter, menu structure for touch-tone IVR systems. These authors determined that reducing the number of menu items per menu to three or fewer does not result in fewer errors. Virzi and Huitema (1997) investigated selection times associated with broad versus deep menu structures for touch-tone IVR applications. Specifically, they tested touch-tone IVR systems with a single, eight-item top level menu against identical systems with the top menu split, with the fifth item in the first set providing access to the final four items (a method described previously). These researchers found that it takes participants longer to make selections when the menu is split in this manner.

Using speech-enabled systems, Vanhouche, Neeley, Mortati, Sloan, and Nass (2001) attempted to determine which prompting style is most appropriate to use with broad menus and which is most appropriate to use with deep menus. They determined that a "delayed" strategy was best for broad menu structures and an "up front" strategy was best for deep structures. The delayed strategy prompts users with an open-ended question, followed by the set of options if the user does not respond within a given period of time or if the response is not in grammar or not recognized. These researchers intentionally placed the target item at the end of the menu list for all trials, which, as discussed above, is not an externally valid method. In a real system, if all items were chosen with equal frequency, then the target item would appear in each position equally as often, rather than always at the end of the list. However, in actual practice, certain targets are chosen with a much higher frequency than others, and therefore are placed at the beginning of the list (a fact that renders their procedure even less valid).

The studies above addressed related topics, but no researcher has yet empirically investigated the effects of implementing single, flat, long menus in comparison to sets of shorter menus with a deeper structure in a speech-enabled IVR system. As discussed previously, speech-enabled user interfaces have different properties than manual key press interfaces; therefore it would be incorrect to attempt to generalize results from any study using a DTMF-based IVR system. Further, no researcher has used signal detection theory to make predictions of user behavior for a speech-enabled IVR or applied MacGregor et al.'s (1986) criterion-based decision model to IVR menu design.

Purpose

The purpose of this dissertation is to investigate user behavior, performance, and satisfaction associated with two IVR menu designs. The information will provide support for or refute hypotheses derived from theories of working memory, search, and signal detection.

PHASE I

Introduction

Background and Purpose

An email-based voice portlet could use either a single long or set of short menus to allow users to access and manage their inbox via phone. The long menu system would be simple, after listening to each mail message, users would be presented with a set of 8 – 11 mail navigation and management options. These options are listed below. Items in brackets may or may not be present in the menu, depending on the context. For example, “Next” would not play when the user has reached the last item in the list and “Previous” would not play when the user is working with the first message in the list.

1. [Next]
2. [Previous]
3. Repeat
4. Delete
5. Reply
6. List recipients
7. [Reply to all]
8. Forward

9. Mark unread
10. Add sender
11. Time and date

This long menu violates the guideline set forth by many that speech IVR menus should be limited to five or fewer items. As described previously, the best way to split a long menu into a set of shorter menus is to reorganize the items into higher-level groups. The purpose of Experiment 1 was to determine the most appropriate way to split the mail menu into separate menus, each containing five or fewer items. Experiment 2 will provide the labels for the new, higher-level menu.

Matching the User's Mental Model

There are a variety of techniques for determining the most appropriate way to group interface items to match the users' mental models. Some of the most popular methods are scaling techniques such as multidimensional scaling (MDS) and cluster analysis. As Eberts (1994) explains, MDS emphasizes the dimensional and spatial structure of the data, and cluster analysis is more appropriate for categorizing data. Cluster analysis is one of the best ways to sort data into logical groups. When conducting a cluster analysis, users provide information about the perceived similarity of items, typically either by rating the similarity of each pair of items (e.g., 1 = extremely similar and 7 = extremely dissimilar) or by sorting items into groups. The data from the similarity ratings or the sorted groups are then statistically analyzed to determine appropriate categorization of items and the implied menu structure.

When participants sort items into groups (rather than providing similarity distances), they often use physical or virtual cards with labels and combine these into groups. This class of approach is referred to as "card sorting." Loshe, Walker, Biolsi, and Rueter (1991) used card sorting to gain a better understanding of users' mental models of a set of graphics. Redish (2005) describes how she used this technique to redesign the US National Cancer Institute's Division of Cancer Prevention website. When conducted manually, card sorting data collection and analysis can be extremely time consuming; however, there are now a variety of automatic card sorting and cluster analysis tools (see Zavod, Rickert, and Brown, 2002 for a review) that make this task much more manageable. Harper, Rhoedenizer-VanDuyne, Jentsch, Smith-Jentsch, and Sanchez (2002) found no differences in results when using manual card sort than when using an automated card sort tool (the TPL-KATS).

Method

Participants

Twenty-six male and female individuals with at least 3 months experience using a corporate or popular free Internet email system (e.g., Yahoo, AOL, Hotmail) participated in this experiment. The majority of the participants were employed at either IBM or Cross Country Trav Corps, each located in Boca Raton, Florida.

Materials

This study employed the automatic card sorting and cluster analysis programs (respectively, Usort and EZCalc) developed by the user-centered design group at IBM. Each program ran on an IBM T41 Thinkpad. The T41p runs Windows XP Professional v1.08, has a 1.7 GHz Intel Pentium M processor, and 1.00 GB of RAM.

Procedure

It was explained to participants that the purpose of the research is to determine the best way to organize menu items for a speech-based e-mail system. Participants read a brief description that clearly explained the action that occurs when each menu item is selected (see Appendix A) and then used the automatic card-sorting tool to place the 11 e-mail menu items into groups of five items or fewer. The Usort card sort tool provides a new, randomized order of item presentation for each participant. Figure 3 depicts the user interface for the card sorting task. Participants were encouraged to ask any questions they had about the sorting task or how to use the Usort user interface to complete the card sort.

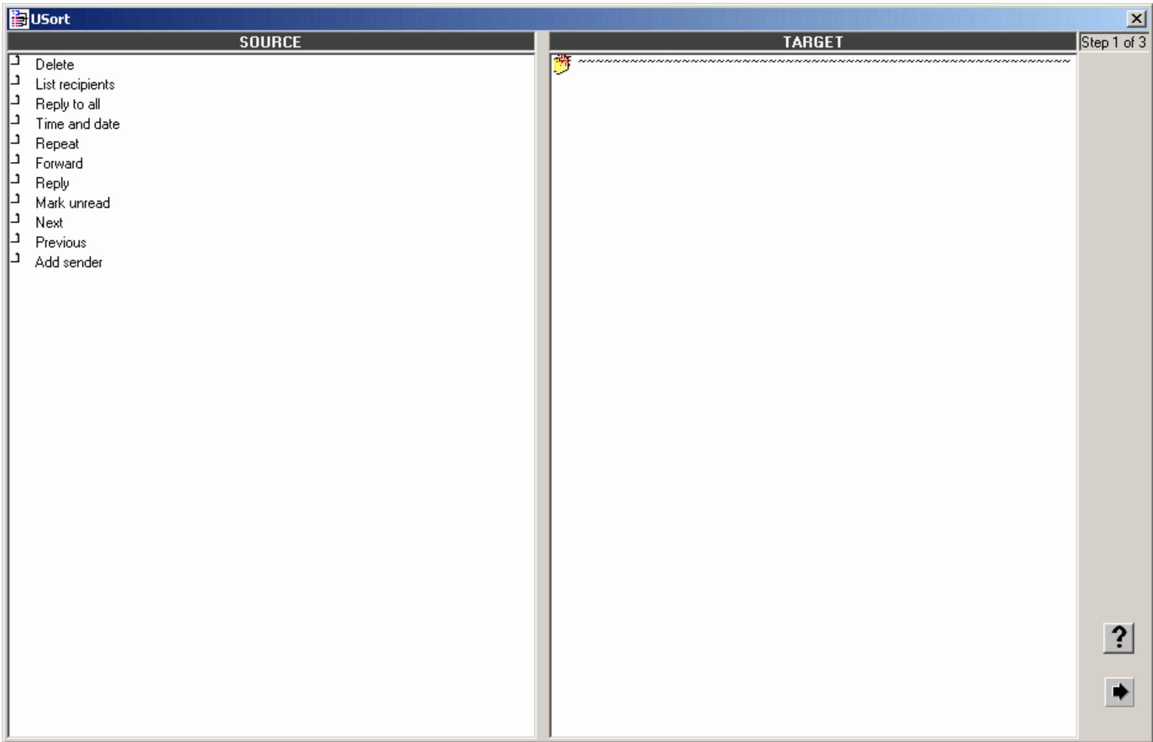

Figure 3. Usort main user interface

Results

The data were analyzed using the average linkage algorithm provided by the EZCalc cluster analysis program. Average linkage is an agglomerative cluster analysis approach by which the distance between clusters is defined as the average distance between all pairs of points (Dillon and Goldstein, 1984). The program provides visual representation of the participants' aggregated mental models produced via the aforementioned analysis of the item distance matrix (see Appendix B). Figure 4 reveals the menu structure, produced by EZCalc, which provides the best match to users' mental models. The output indicates that the menu should be comprised of four groups: 1) Delete, Forward, Reply, and Reply to All; 2) Repeat, Next, and Previous; 3) Mark Unread and Time and Date, and 4) List Recipients and Add sender.

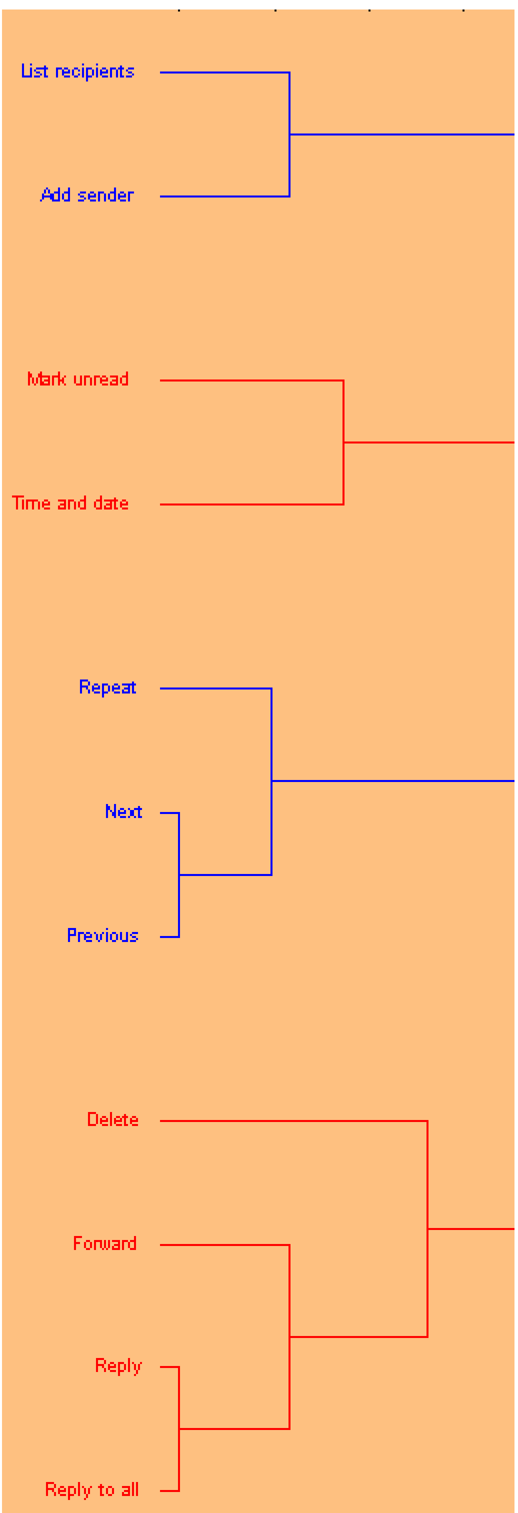

Figure 4. Cluster analysis output for e-mail system menu

PHASE II

Introduction

Background

Experiment 1 indicates that the best way to organize a speech-enabled e-mail system into a deeper structure with shorter menus is to group the eleven options into four higher-level menus. The next step is to determine the most appropriate labels to appear in the new higher-level menu. To accomplish this, a series of two Internet-based user surveys were conducted. The first survey produced the label candidates for each group and the second determined the winning labels.

Method

Participants

An e-mail message invited 1000 male and female IBM employees (all Lotus Notes users) to participate in Survey 1 and a separate group of 1000 employees to participate in Survey 2. The response rate for Survey 1 was 10.1% (101 participants) and the response rate for Survey 2 was 15.5% (155 participants).

Materials

The web-based surveys were created and administered using WebSurveyor Desktop v4.0 software. Participants accessed the surveys via the web browser of their choice.

Survey 1

Procedure. Participants read a description of the function of each of the menu items (see Appendix A), examined each menu item group, and suggested a label for each of the menus. When satisfied with their responses, participants clicked the "submit" button. The survey was available for five days.

Results. A total of 101 participants provided suggested labels for each of the menu groups. For the group containing Next, Repeat, and Previous, 22 respondents suggested "Navigate," "Navigation" or a closely-related term with "Navigate" (12 respondents) being the most common suggestion of this group. Fifteen respondents suggested "Listen to messages" "Listening to messages" or a closely-related phrase with "Listen to messages" (7 respondents) being the most common of the group. There were no other response groups in which more than four participants suggested similar items; therefore, "Navigate" and "Listen to messages" were selected as the top two candidate labels for this group.

For the group containing Reply, Reply to All, Forward, and Delete, 22 participants suggested “Action,” “Actions,” “Message actions,” or something closely related, with the most common suggestion being, “Actions” (7 respondents). Nineteen participants suggested “Respond,” “Response,” “Responding” or something closely related, with the most common suggestion being, “Respond” (6 respondents). No other label was suggested more than three times; therefore “Actions” and “Respond” were selected as the top two candidate labels for this group.

For the group containing Add Sender and List Recipients, 25 participants suggested “Address,” “Address book” or a closely-related word or phrase, with the most common suggestion being “Address book” (9 respondents). Nine participants suggested “Distribution” or “Distribution List,” with “Distribution” being the most common suggestion of this group (6 respondents). No other item was suggested more than twice; therefore “Address Book” and “Distribution” were selected as the top two candidates for this menu label.

For the group containing Mark Unread and Time and Date, 10 participants suggested “Message Details” “Message Information” or a related phrase, with “Message Details” being the most commonly-suggested label for this group (5 respondents). Ten respondents also suggested “Status,” “Message Status” or something closely related, with “Status” being the most commonly-suggested label for this group (5 respondents). Five participants suggested “Miscellaneous” and five suggested “Options.” No other term or phrase was suggested more than twice. “Message Details,” “Status,” “Miscellaneous,” and “Options” were selected as the four top candidates.

Survey 2

Procedure. For the second survey, 1000 participants were invited to participate in one of two surveys with the only difference being the order in which the options were presented. In one version, for each menu set, the label options were presented in alphabetical order. The other version presented the label options in reverse alphabetical order. Participants read the description of the menu items (see Appendix A), examined each menu item group, and selected, via multiple choice format, the most appropriate label from the list of selections generated in Survey 1. Participants completed the survey by clicking the submit button. The survey was available for five days.

Results. There were 155 total respondents. Seventy-three participants responded to the alphabetical version and eighty-two responded to the reverse alphabetical version. For the group of menu items that contained Next, Repeat, & Previous, 130 participants (83%) selected “Listen to Messages” and 25 selected “Navigate” (17%). A chi-square analysis revealed that this distribution differs significantly from that which would be expected by chance ($X^2(1) = 19.593$; $p < .0005$). For the group that contained Reply, Reply to All, Forward, and Delete, 102 participants (66%) selected “Respond” and 53 (34%) selected “Action.” A chi-square analysis revealed that this response distribution also differs from that which would be expected by chance to a significant degree ($X^2(1) = 15.49$; $p < .0005$). For the group that contained Add Sender and List Recipients, 99 participants (64%) selected “Distribution” and 56 (36%) selected “Address Book.” This difference was also statistically significant ($X^2(1) = 11.93$; $p = .001$). For the menu group containing Mark Unread and Time and Date, 62 participants (40%) selected “Message Details,” 40 (26%) selected “Options,” 38 (25%) selected “Status,” and 15 (10%) selected “Miscellaneous.” This distribution was also significantly differently than that which would be expected by chance ($X^2(3) = 26.39$; $p < .0005$). Visual examination of the cells (as described by Gardner, 2001) suggests that the significant differences are driven by the residuals of “Message Details” (+23.3) and “Miscellaneous” (-21.8). Based on these results, “Listen to Messages,” “Respond,” “Distribution,” and “Message Details,” were selected as the four most appropriate menu labels.

Discussion

The sum of the data collected in Phases 1 and 2 lead to the deep structure, short menu item system that would most closely match users’ mental models and expectations, given the set of eleven commands that are available after listening to a message in a speech-enabled mail system. The design should contain four menus labeled “Listen to Messages” (Next, Previous, and Repeat), “Respond” (Reply, Reply to All, Forward, and Delete), “Distribution” (Add Sender and List Recipients), and “Message Details” (Mark Unread and Time and Date). Figure 5 illustrates the general design.

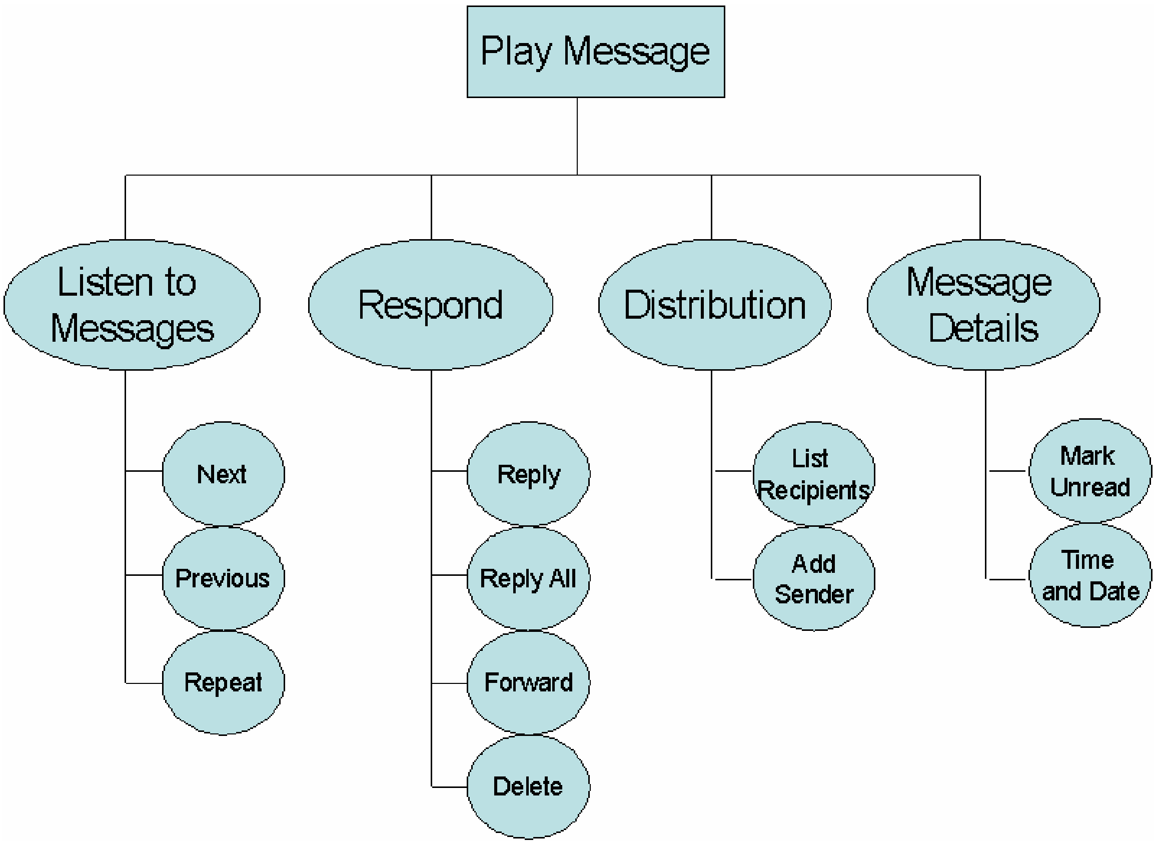

Figure 5. Deep, short menu IVR design

PHASE III

Introduction

Background

Currently, a commonly-held belief within the IVR design community is that designers should keep menus to a small number of items (most advocating five or fewer). The reason for this assertion is that users are typically unable to immediately recall more than 5-9 items – the assumption being that the inability to immediately recall all menu items will lead to errors. However, this paper has argued that the opposite is true – that artificially distributing menu items that hang well together over a set of menus and providing access through higher level menu(s) will lead to a greater number of errors and time-on-task. Modern theories of working memory, search behavior, and signal detection support these hypotheses. Further, based on theories of search and signal detection, several hypotheses have been made about the search behaviors that each design is likely to lead to, the errors associated with these behaviors, and the consequences of these errors.

As previously discussed, the most appropriate method of shortening a menu to comply with the guideline such that menus should not be greater than five items in length is to employ user-centered design methodologies to organize menu items into smaller, related groups and provide a higher-level menu from which users can access these new menu groups. The goal of Phases 1 and 2 of this dissertation was to determine the most appropriate way to design a phone-based e-mail system, using a set of eleven menu items, but with no individual menu containing more than five items. The card sort and cluster analysis conducted during Phase 1 provided information as to the best way to organize e-mail menu items into smaller groups that match email users' mental models and the two surveys conducted during Phase 2 provided the best higher-level menu labels for these new groups.

Purpose

The purpose of this experiment is to provide support for one of the two aforementioned theories regarding the cognitive processes involved in speech-enabled IVR use. The traditional view will find support if users perform poorer with and are less satisfied with the single long menu phone-based e-mail system. If there is an interaction such that this effect is more pronounced for low working memory capacity users, this will lend further support to the traditional view. On the other hand, my analyses will be supported if users have more difficulty with a voice system that uses a set of shorter menus and a deeper structure. Results of this study will also provide information as to the validity of my arguments regarding the search behaviors that users will engage in based on the menu structure. The results of this experiment will have strong implications for menu design, particularly in the realm of speech-enabled telephony systems.

Hypotheses

1. Users will perform better (in terms of time-on-task and task success) and will be more satisfied, with the single long menu e-mail voice system than the deeper, short menu system.
2. Low working memory capacity participants will perform worse and will be less satisfied than high capacity participants.
3. There will be interactions such that low working memory capacity participants' performance and satisfaction levels will be more negatively affected by short menu structure.
4. Users of the deeper menu system will engage in a greater proportion of self-terminating searches than users of the single long menu system.
5. Users of the deeper menu system will engage in riskier behavior, achieving a greater proportion of hits and smaller proportion of misses, but also committing more false alarms and failing to correctly reject more items than users of the single long menu system.
6. Mean response time will be shorter for correct selections than for incorrect selections.

Method

Participants

A total of 121 participants completed working memory capacity screening and 58 participants were retained for the IVR study. All participants were undergraduate students at the University of Central Florida with at least three months experience using corporate or web email.

The mean participant age for those completing the main experiment was 20.3 years and the range was 18 - 40. All participants received course credit for their participation in the study. No participants reported having a speech or hearing deficit and no participants reported having experience designing or programming speech applications.

Materials

This experiment employed two versions of an e-mail voice application. Each of these voice applications were pre-populated with a set of seven email messages for which participants would access and act upon. Each IVR played each mail message and then offered 8 – 11 email-related menu options, depending on the message position ("Next" or "Previous" could be omitted) and number of recipients ("Reply All" could be omitted). The long, broad version played all options immediately following each message in a single menu (Figure 6 illustrates this general design). The short, deep version split the menu items into four menus, which could be accessed through a higher-level menu that is played immediately following a message (see Figure 5). From this point on, these two IVRs will be referred to as the "Long" and "Short" versions, respectfully. Each system employed the IBM WebSphere Voice Server v3.1 U.S. English Female Super Voice (a concatenative text-to-speech voice). Appendix C provides the VXML 1.0 code for each of these IVRs.

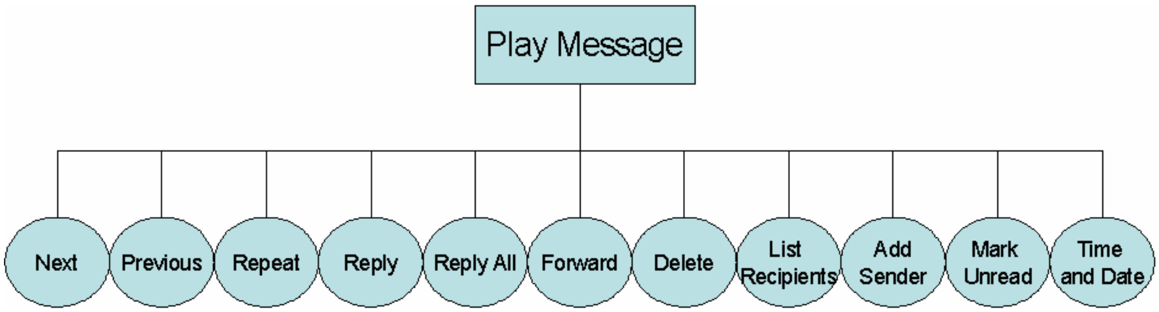

Figure 6. Long, broad menu IVR design.

Participants also completed the automated operation span task created by Unsworth, Heitz, Schrock, and Engle (2003). The authors demonstrated that this version of the operation span test is reliable and valid, correlating moderately with Turner and Engle's (1989) operation span test and showing high test-retest reliability. The test presents participants with an arithmetic operation (e.g., 2*3 – 1) followed by a panel with an answer to the equation (e.g., 7) that may be correct or incorrect. The participant selects True or False from the computer monitor and then the system presents a letter to be retained for recall at the end of the trial. The participant answers a series of 3 - 7 equations followed by letters and is then prompted to recall the letters in the order they were presented. The test consists of a total of 75 math problem/letter combinations. Participants learn and practice the math operations and letter recall tasks separately and then together before proceeding to the actual tasks. The automated operation span score is the total number of letters that were recalled in their respective position throughout all trials. For example, if a participant recalls 3 of 3 in order for one set and 4 of 6 in another, that participant's score would be 7 (3 + 4).

This experiment also employed the revised version of the Post-Study System Usability Questionnaire (PSSUQ; Lewis, 2002), a subjective usability questionnaire with a history of established psychometric quality (Lewis, 1995; 2002). The PSSUQ contains 16 items for which participants indicate their level of agreement on a 7-point scale, with lower ratings indicating greater subjective usability and user satisfaction (see Appendix D). The PSSUQ has three subscales, which correspond to the general areas of system usefulness, information quality, and interface quality. The PSSUQ Score is the arithmetic mean of the 16 ratings.

The experimenter used Sony Sound Forge v7.0 and a phone tap to record and save the experimental sessions electronically. The IVR system was hosted at the IBM facility in Boca Raton, Florida and participants interacted with the system using a regular telephone handset. A Radio Shack volume booster was used to increase the volume for recording purposes and was connected to a speaker, which allowed the experimenter to hear the IVR system.

Experimental Design

The experimental design included two between-subjects independent variables, each with two levels (Menu Design: long, short; Working Memory Capacity: low, high), and one within-subjects variable (Response Correctness). The experimenter assigned each participant to either the Short or Long menu group and to one of four task orders via a randomization process with constraints. These assignments were made by following a predetermined sequence laid out on a "condition tracking sheet." This procedure set to ensure that approximately the same number of participants from each WMC group would use the Short system as the Long system. This method of randomly assigning, with constraints, participants to groups should ensure appropriate distribution of all other individual differences (e.g., ability to learn; familiarity with technology) present in the participant pool (Cook & Campbell, 1979). The method of assignment also set to ensure that approximately the same number in each group worked with each of four task sequences. The task sequences were creating using a random number table (Appendix E provides the four task orders). Each of these tasks required that users interact with either a single, long menu or a set of shorter menus, depending on group assignment.

Specific Hypotheses

1. Participants who use the Long system will complete a significantly greater number of the tasks than those who use the Short system
2. The total time to complete all tasks will be significantly greater for Short system participants than for Long system participants
3. Participants using the Long system will indicate significantly higher levels of satisfaction (lower PSSUQ scores) than those using the Short system
4. Participants with low working memory capacity will complete significantly fewer tasks than those with high working memory capacity
5. The total time to complete all tasks will be significantly higher for participants with low working memory capacity than for those with high working memory capacity.

6. Participants with low working memory capacity will indicate significantly lower levels of satisfaction (higher PSSUQ scores) than those with high working memory capacity.
7. There will be an interaction such that the users will complete more tasks with the Long menu system and this will be true to a significantly greater degree for those with low working memory capacity
8. There will be an interaction such that it will take users less time to complete tasks with the Long menu system and this will be true to a significantly greater degree for those with low working memory capacity
9. There will be an interaction such that users of the Long menu system will provide lower PSSUQ ratings (indicating greater satisfaction) and this will be true to a significantly greater degree for those with low working memory capacity.
10. Users of the deeper menu system will engage in a greater proportion of self-terminating searches than users of the single long menu system.
11. Users of the deeper menu system will achieve a greater proportion of hits
12. Users of the deeper menu system will commit a greater proportion of false alarms
13. Mean response time will be shorter for correct selections than for incorrect selections.

Procedure

First, participants read and signed the informed consent form (see Appendix F) and then completed the background questionnaire (Appendix G). All qualifying participants then completed the automated operation span task. The automated operation span task worked as a screening tool (one of its intended purposes) to eliminate those participants whose score fell into the middle third of the expected distribution. The expected distribution was calculated based on data collected by Unsworth et al. (2003) with a student population from another major US university and individuals in the community surrounding the university. The Unsworth, et al. data were normally distributed and the paper provided the mean, standard deviation, and upper and lower quartiles, from which the upper and lower thirds were calculated. Based on this calculation, all participants whose automated operation span score fell between 33 and 46 were dismissed. All participants who committed more than 20% math errors were also dismissed, since it was likely that they spent too much resource rehearsing letters at the expense of the operations tasks. In total, 121 participants completed the working memory capacity test and 80 continued with the remainder of the experiment. Participants who scored within the selected operation span ranges were assigned, based on the condition-tracking sheet, to either the Long or Short menu version of the IVR and to one of the four task orders. This procedure helped to obtain an approximately equal number of data points from each of the four cells.

Once qualifying participants completed the working memory span test (those with middle-third WMC scores were excused), they began the IVR tasks. The experimenter first provided participants with a password (Gold or Black), which activated the appropriate IVR and the experimenter explained to participants that they would attempt to accomplish a set of mail management tasks using an automated speech-enabled phone application. The experimenter further instructed that they would have a maximum of five minutes to complete each task. Prior to beginning each task, participants read the task (see Appendix E for a list of all tasks) and asked any questions they had about the task to be next attempted. Some of these tasks required users to obtain information (e.g., the time a message was sent), some required users to complete a transaction (e.g., reply to a message), and one required them to obtain information and use that information to complete a transaction (e.g., find out what a sender wants and then follow through with the request). Once participants indicated that they understood the upcoming task, the experimenter instructed them to use the phone number provided to access their inbox via the IVR. The task sets required users to utilize all eleven of the menu items at least once to successfully complete all tasks (with the exception that perfect performance on all tasks could prevent the necessity to use "Previous" and "Repeat"). Therefore, to complete these tasks, users had to make early, middle, and late selections from the Long menu and selections from all positions within each menu of the Short menu.

Once users indicated that they had completed each task or once the five minute time interval had elapsed, the experimenter instructed them to hang up, ended the recording, and instructed the participant to move on to the next task. Upon completion of all tasks, the experimenter administered the PSSUQ. The recordings were saved for future analyses pending a final working memory screening procedure to determine the upper- and lower-quartiles for my data set (described below). The initial WMC screening, based on the upper- and lower-thirds of Unsworth et al.'s (2003) data allowed dismissal of participants who clearly would not be included in the final data set. After determining the final set of high- and low-WMC participants (through a procedure described below), the data were analyzed to obtain task completion time, task success rate, SDT outcomes, search strategy, and response time measures.

Results

Working Memory Capacity

Five participants were removed from the data set before beginning analyses. Two participants scored a zero on the automated operation span test, which indicates that they did not attempt to do well and three participants committed too many math errors, which indicates that they allocated too much resource to the rehearsal of letters; therefore there is no way to know the these participants' true scores. After removal of these scores, the final data set included 116 scores. The mean score was 42.3, the median was 42.5, and the standard deviation was 14.96. The top quartile included all participants who scored 54 and above and the bottom quartile included participants with scores below 33 and one of four participants who scored 33 (chosen at random). From this set, there was one participant in each group who had been dismissed from the experiment for reasons other than operation span score. Each participant was replaced by selecting one of the participants with the next score (33 for the low-WMC group and 52 for the high-WMC group). Each participant was chosen based on the conditions in which they had participated. For example, since there were 15 Short menu participants and 13 Long menu participants, one of the participants who scored 33 on the WMC test and used the Long menu system was selected. This participant also had used a task order which had lower representation than the other participant who used the Long system and scored 33 on the WMC test.

The procedure described above provided 29 low- and 29 high-WMC participants and 31 Short system users and 27 long system users. Table 1 provides the number of participants in each of the four cells.

Table 1. Distribution of participants among the four conditions

Menu Design	Working Memory Capacity	
	High	Low
Long	13	14
Short	16	15

Performance and Satisfaction

A set of three two-factor analyses of variance (ANOVAs) were conducted. This experiment employed the set of two-factor ANOVAs as opposed to a MANOVA for two reasons. First, for this experimented, there is no value of assessing the effects of all DVs aggregated as a single unit as this analysis would not support or refute any of the stated hypotheses. The second reason to consider a MANOVA would be for use as a gateway to the ANOVAs to avoid committing a Type I error. However, Abelson (1995) and Wilkinson & APA Task Force on Statistical Inference (1999) both advise that this is overly stringent and unnecessary for cases when the researcher tests a relatively small number of effects and articulates theoretically-founded a priori hypotheses of the expectations (as is the case for these analyses).

The independent variables were Menu and WMC, and the dependent variables were Total Time (total time to complete all tasks), Complete (number of tasks successfully completed) and PSSUQ score. The analyses indicated that there was a main effect of Menu for Total Time, $F(1,54) = 67.551$, $p < .0005$, such that it took participants significantly longer to complete all tasks when using the Short system (M = 1358 seconds) than the Long system (M = 917 seconds). There was also a main effect of Menu for Complete, $F(1,54) = 35.142$, $p < .0005$, such that those using the Short system completed significantly fewer (M = 5.26) tasks than those using the Long system (M = 6.70). The ANOVA revealed a final main effect of Menu on PSSUQ, $F(1,54) = 19.850$, $p < .0005$, such that those using the Short system indicated that they were significantly less satisfied with the system (M = 4.17) than those using the Long system (M = 2.64).

There was a main effect of WMC such that those with high working memory capacity completed significantly, $F(1,54) = 4.223$; $p = .045$, more tasks (M = 6.17) than low-WMC participants (M = 5.69). There were no significant differences between high- and low-WMC users in terms of time to complete all tasks or satisfaction. The ANOVA revealed a significant interaction between WMC and Menu, $F(1,54) = 4.536$, $p = .038$, for Total Time. The interaction was such that participants with high WMC completed tasks only slightly faster than those with low WMC when using the Long system (M_{diff} = 54 seconds; 7.7 seconds per task), but were significantly faster when using the Short system (M_{diff} = 172 seconds; 24.6 seconds per task). Figure 7 depicts this interaction. There was not a significant interaction for task completion rate or subjective ratings.

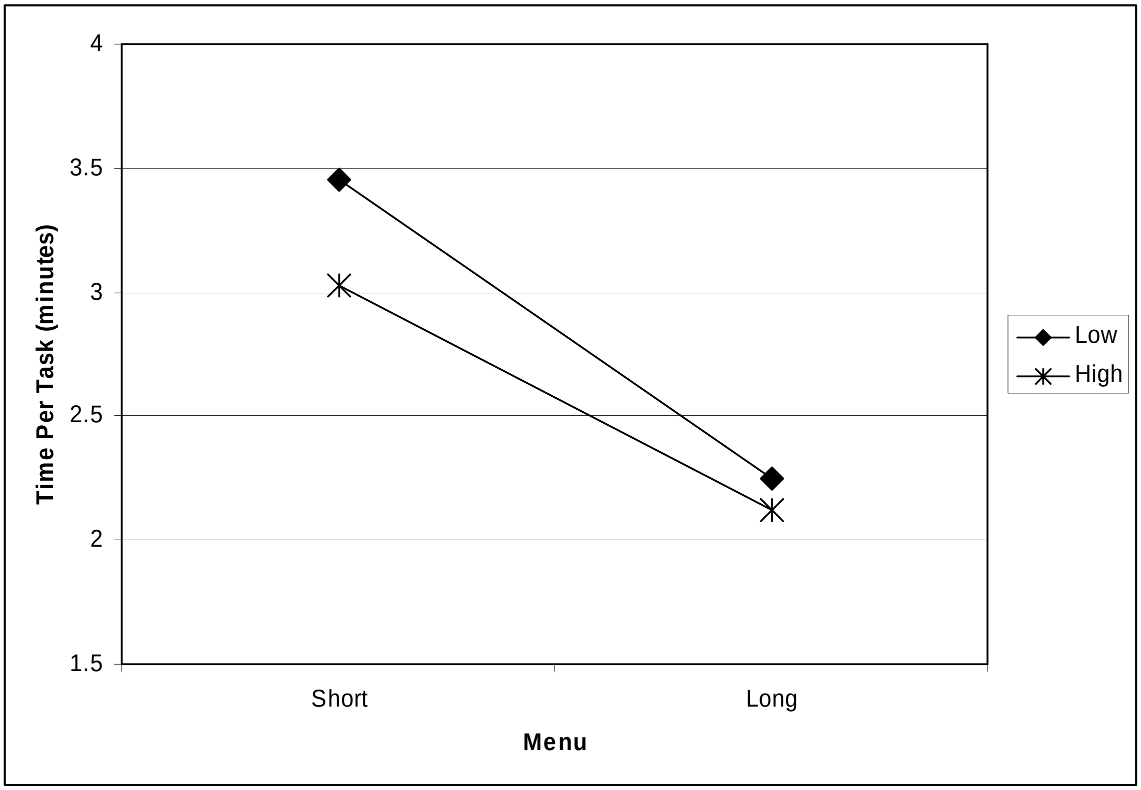

Figure 7. Time per task for high- and low-WMC participants using each system.

Search and Signal Detection

A detailed auditory analysis of tasks 1, 4, and 7 for all participants was conducted. This provides a good sampling of participants' behavior on the initial trial, after some learning has occurred, and on the final trial. A set of three single factor ANOVAs determined whether there were significant differences between those who used the Short system and those who used the Long system in terms of proportion of self-terminating searches, hits, and/or false alarms. The following sections describe the results of statistical analyses performed using data from these three tasks.

Search behavior. All searches were classified as self-terminating, exhaustive, or redundant based on the descriptions of these types of searches provided earlier. All searches were defined by the position in the menu at which the participant first attempted to give a voice command. If the participant gave a voice command before hearing it, this was still coded as a self-terminating search. When using the Short system, the coding was based on the point within all menu items the participant had reached when the participant first issued a command, excluding navigational commands. For example, if a participant chose “Listen to Messages” and then chose “Next” after listening to all three options within this sub-menu, the item was a self-terminating search, because the user terminated the search before presentation of all eleven menu items. However, if a user re-reviewed one or more sub-menu items and then made a selection (without having been presented with all items), it was coded as a redundant search. The proportion of self-terminating searches was determined by dividing the number of self-terminating searches by the total number of searches.

The ANOVA indicated a main effect for Menu, $F(1,56) = 21.568$; $p < .0005$, such that Short system users engaged in significantly greater proportion of self-terminating searches (M = .979) than Long system users (M = .841).

Signal detection outcomes. Table 2 provides the signal detection outcomes for participants who used each of the two IVR systems. As the IVR presented each menu item, participants could select the item or reject it (by not selecting it). When participants selected a target item within 2 seconds of its presentation, it was coded as a hit; if the participant did not select the item within the 2-second window (or if they did not select the item at all), it was coded as a miss. When participants selected a distracter item within 2 seconds of its presentation, it was coded as a false alarm. When they rejected a distracter item, by choosing not to select it within 2 seconds of presentation, it was coded as a miss.

Table 2. Proportion of each signal detection outcome for participants using each system

	Hit	Miss	Correct Rejection	False Alarm
Long	.629	.371	. 984	.017
Short	.607	.393	. 948	.052

The ANOVA indicated that Short system participants committed a significantly, $F(1,56) = 9.462$; $p = .003$, higher proportion of false alarms ($M = .052$) than Long system users ($M = .016$). The difference between the proportion of hits for the Short and Long system users was not significant.

Since many of the individual participants had hit rates of 1 and/or false alarm rates of 0, it was not possible to calculate individual z scores for these participants and therefore unable to make subsequent comparisons of group means for *c* and *d'*. However, using the group mean proportions of hits and false alarms for the Short and Long menus, and applying formulae provided by Macmillan and Creelman (2005), the response criterion, *c*, and sensitivity, *d'*, scores were calculated for each group. Table 3 provides this information. Higher *d'* scores indicate a greater sensitivity. Positive criterion scores indicate a bias to reject items (a negative bias), the higher the score, the greater the bias.

Table 3. Sensitivity (d') and response criterion (c) scores for each menu design

	Short	Long
Criterion (*c*)	.678	.905
Sensitivity (*d'*)	1.90	2.47

Response time. To determine whether there were significant differences between the speed with which participants provided correct and incorrect responses, a paired samples t-test was conducted. The t-test included only the 39 participants who committed at least one incorrect response (19 did not provide an incorrect responses). Mean time to correct response was 2.84 seconds and mean time to incorrect responses was 3.12. The paired-samples t-test did not indicate a significant difference between the means of these correct and incorrect responses, $t(38) = 0.807$; $p = .425$.

Discussion

Working Memory, Performance, and Satisfaction

Citing Miller (1956), many authors and design experts warn that IVR menus should never contain more than five items. The reasoning is that, since most individuals can serially recall an average of 5 – 9 newly-presented items, menus containing greater than 5 items will tax users' memories. The implied assumption is that it is necessary to remember all menu items in order to effectively utilize an IVR. This paper has argued that it is not necessary for users to retain all menu items to work efficiently with an IVR and that IVR use is largely an auditory search task.

Modern theories (e.g., Baddeley and Hitch, 1974; Just and Carpenter, 1992) view working memory as more than a simple, temporary information storage unit. Instead, working memory is now theorized to be responsible not only for storage, but also for advanced activities, such as attentional control and operating on stored information (for example, performing mathematical computations). Based on these modern theories of working memory, this paper proposed a model of IVR use (see Figure 2) such that individuals hold one or two top candidates in working memory while evaluating each newly-presented item. Individuals then weigh the newly-presented item against the current candidate(s). This evaluation results in a decision process to either replace the current top candidate(s) with the newly-presented item or to discard the new item and maintain the established top candidate(s). The model proposes that, regardless of menu length, users need only store one or two items, while evaluating another. This paper further argued that when a group of items are all applicable at a particular point in an IVR system, splitting these items and creating a deeper menu structure will demand additional working memory resource. The primary reason for this is that it requires users to store navigational information and engage in way-finding processes while storing and evaluating IVR items.

The results of this experiment support my hypotheses, as participants who used the Long menu structure significantly outperformed participants who used the Short structure and indicated significantly higher levels of satisfaction with the user interface. Further participants with low WMC were more negatively affected by the Short menu system than participants with high WMC. These findings suggest that short, deep menu structures, rather than flat, long menu structures are actually more demanding on users' working memory resource.

The finding that the Long system afforded greater performance and satisfaction suggests that it is less demanding and easier to interact with. In addition to reduced strain on working memory capacity, there are likely other attributes of an IVR system that has been artificially deepened and shortened that are detrimental to performance. For example, increasing depth increases the number of speech utterances a user must provide to navigate the IVR. Increasing speech turns will necessarily increase time-on-task, even when performance is perfect. However, neither the user's nor the speech system's performance is perfect. Therefore, increasing user-system turns also increases the number of errors that each commits in a given session. Also, adding additional layers requires that participants engage in additional class-inclusion searches. In this experiment, participants recognized most of the target commands in the Long system, but often had trouble identifying which higher-level menu would provide access to these commands. An excellent example of this is that participants using the Long system always immediately recognized "Next" as the target command to move to the next message in the list; however, it was not so obvious to users of the Short system that they should choose "Listen to messages" to access a command which gives them the ability to move forward to the next message.

Considering that many arguments can be made for why individuals using the Long system outperformed individuals using the Short system, the evidence that individuals were affected differentially based on their working memory capacity is particularly compelling. High-WMC participants of the Long system did not significantly outperform low-WMC users (a difference of only 7.7 seconds per task); however, high-WMC participants using the Short system did significantly outperform low-WMC participants (a difference of only 24.6 seconds per task). This provides strong evidence that the Short menu significantly taxes one's working memory – one of the factors contributing to its overall poor usability and contributing more substantially to the detriment of those with low WMC. In other words, if one wants to design a system that is easier to use for all, a single flat menu is the best choice. If instead one wants to design a system that is more difficult to use, particularly for those with weaker WMC, this can be accomplished by splitting menu items into a deeper structure.

Given modern theories of working memory capacity and a detailed analysis of the IVR use, these results are not surprising. True, modern theories of working memory do not refute that capacity is limited, but an analysis of the task reveals that there is no need to retain all menu items in one's working memory; nor is the ability to serially recall all items necessary. Instead, modern theories of working memory provide insight as to the general activities that are demanding of working memory resource. When using a single long menu, it is only necessary that the user has the ability to hold one or two items in working memory while perceiving and evaluating a comparison item. However, when user a shorter, deeper structure, individuals must retain much more information and engage in much more effortful processes as well. Users must engage in decision and selection processes at higher-level menus, retain one or two candidate(s) from the lower-level menu, engage in additional navigation processes, retain this navigation information, etc. This type of analysis clearly leads to the prediction that a deeper structure will be more taxing on one's working memory resources. This prediction is now supported with empirical data.

Search Behavior

MacGregor, Lee, and Lam (1986) described the dual-criterion model of search behavior and applied this model to visual search tasks. They found that the fewer items in a menu, the more likely a user is to conduct a self-terminating search. There are two reasons for this. The first explanation is that a greater proportion of target items will appear as the final menu item when the menu contains fewer items. Therefore, if the user selects the target immediately after presentation, this will culminate in a self-terminating search more often for short menus. The second rationale is that that, if a user knows or has estimation that the menu is short, the user will be more willing to review all items before making a selection. This is true because the full review is less effortful and time-consuming under these circumstances.

For the search analyses, the full menu was conceptualized as a list of the 9 – 11 items, which are spread on several auditory "pages" in the Short system and contained in a single menu with the Long system (just as Snowberry et al., 1983 spread multiple items across visual pages and compared this to menu items displayed on a single page). Based on the second argument presented above, it was hypothesized that users of the Short menu system would engage in a greater proportion of self-terminating searches than Long system users. In other words, the expectation was that the increased effort necessary to review all menu items (in terms of time and need to navigate) would deter users from engaging in this full review.

As predicted, the Short system users engaged in a significantly higher proportion of self-terminating searches than Long system users. Both Short and Long users generally encountered the target item prior to the final item's presentation, but Long system users were more willing to review the remaining items before making a selection. Likewise, Short system users seemed more willing to make a selection for which they were not fully confident to avoid continued search. It should be noted, that this can also partially be explained by the fact that the last item in the Long menu structure ("Mark Unread") is only the final item for the Short system if users choose the incorrect higher-level menu several times. However, this likely only contributed slightly to a finding that can largely be explained by Short system participants perceiving the costs of a full review as more substantial than Long system participants. For example, it was common for Short system users to make incorrect selections after entering a sub-menu, rather than to back out of the menu and continue searching for a more appropriate menu item.

As a final note on this topic, Short system users did engage in a substantial proportion of exhaustive and redundant searches of the four-item, higher level menu (this menu was excluded from analysis). This finding is consistent with MacGregor, Lee, and Lam's (1986) arguments given that 25% of the items appear at the end of the menu and that the effort to review four items is small.

Signal Detection Outcomes

The two main factors that typically influence response bias are probability expectations and payoffs, with previous research indicating that individuals modify their response criterion to a greater degree based on payoffs than on probability of signal (Bisseret, 1981; Harris and Chaney, 1969 as cited in Wickens & Hollands, 2000). In this experiment, participants knew that all tasks were possible. In other words, for each menu, there would always be a single target item and 7 – 10 distracter items. Based on this, probability of target presentation was not expected to influence response bias differentially for the Short and Long systems. On the other hand, the payoffs associated with each of the four signal detection outcomes were such that the theory would predict riskier behavior for those using the Short system. Based on these consequences, it was predicted that those using the Short system would achieve a greater proportion of hits at the expense of committing a greater proportion of false alarms.

As hypothesized, the results of this study indicate that Short system participants committed a significantly greater proportion of false alarms than Long system participants, with non-statistically significant differences in the hit rate. It appears that the consequences associated with search complexity of the Short system led these system participants to incorrectly select distracters more often, but did not make these participants more likely to select a target item quickly following presentation. When one's response criterion shifts, this will typically result in either an increase or a decrease in both hit rate and false alarm rate. However, in this case, only false alarm rate was higher for the Short system participants.

A potential explanation for these findings is that the differences in false alarm rate with no difference in hit rate between the two groups represent two effects of the Short system: a) a weaker sensitivity and b) a lower criterion. It may be the case that the more severe consequences associated with the misses for the Short system users actually did increase hit rate (in addition to false alarm rate), but that these effects were undetectable due to the counter effects of system, which provided additional, difficult-to-detect target items. This explanation could certainly be supported by the search behavior data described in the previous section such that Short system participants demonstrated riskier behavior - much more commonly making a selection before reviewing all list items than Long system participants.

Speed of Response

There was not a significant difference between the speed with which participants delivered correct and incorrect responses. This may be partially due to differences in participants' response style and understanding of the system. Some participants very quickly picked up on the fact that the system allowed them to "barge in" at any time with a response and other participants did not pick up on this at any point. Therefore many participants chose to wait until all options had been presented before making a selection…correct or incorrect. It is very likely that participant interaction style and system knowledge accounted for more of the variance associated with time to respond than the true speed with which participants would be capable of delivering each response.

Related to this, participants who had learned or could make educated guesses as to the coming target items had the ability to barge in and make a selection before it was provided. These pre-selections were not included in the analyses of time to respond since participants were not actually responding to the presentation of an item. It can be assumed, however, that in these cases, participants would likely select the item very quickly after presentation if preselections were not possible. Since participants preselected target items much more commonly than they preselected distracters, this ability likely had the effect of slowing the mean overall speed of correct responses.

Potential Limitations to this Research

User-centered methodologies were employed to the design of the Short menu system. Domain experts provided information via a card sorting task as to how the individual menu items should be grouped into sets of five or fewer and provided the best labels for the higher-level menu items that provide access to these groups. This makes it very difficult for one to argue that there are alternate and better ways to design a similar phone-based e-mail system with the same set of features and functions, using menus that contain no more than five items. However, one might argue that an e-mail system is a particularly rare domain in which a violation of the "five or fewer" rule results in better design. Therefore, it would be interesting to replicate this experiment with other domains (e.g., banking transactions, phone-based shopping, etc.). It would also be interesting to modify a system with a short menu structure that is known to be highly usable and flatten it, using best design practices, and then compare performance and satisfaction for a group of representative tasks.

All participants in this experiment were undergraduate students from the same university; therefore these results can only be strictly generalized to this population. Future research could determine if the same effects are realized with other populations (e.g., the elderly, those without a high-school degree, etc.). Also, since all participants were drawn from the same university, it is appropriate to assume that the range of working memory capacity scores is less variable than that which would be expected from the entire potential user population of an IVR system. If practical, it would also be interesting to replicate this experiment with a population for which greater variance in working memory capacity would be expected.

Conclusion

This experiment provided evidence that, contrary to common belief, it can be advantageous to design an IVR system to utilize a flatter structure with fewer long menus as opposed to a deeper structure with a greater number of shorter menus. These findings are consistent with predictions based on a thorough examination of modern theories of working memory and detailed analyses of phone-based tasks. The experiment further provided evidence that intensive demand on working memory resource is one of the contributing factors to the performance detriment associated with a design that employs a hierarchical set of menus containing five or fewer items. This argument is supported by the interaction such that low-WMC participants expended similar amounts of time as high-WMC participants when using the Long menu system, but expended significantly more time than high-WMC participants when using the Short menu system. It would be interesting to also manipulate task demand and see if the same type of interactions occurred. In other words, will a short, deep menu structure be differentially more detrimental to performance of more resource-intensive tasks?

As predicted, participants engaged in more self-terminating searches when using the short menu system. This is most likely due to participants' different perceptions of the costs associated with thorough review all menu items. In other words, Long participants were much more willing to review all list items as the necessary effort to do so was substantially lower. These findings provide evidence that MacGregor, Lee, and Lam's (1986) dual-criterion decision model can extend to auditory search. Related, SDT leads to predictions that Short system participants would engage in riskier behavior achieving more hits at the expense of committing more false alarms then Long system users. This was partially borne out in the results as Short system participants committed more false alarms, with an equal number of hits. It's possible that this is due to the combined effects of weakened signal of the Short system as compared to the long combined with riskier behavior for participant's using this system.

The results of this set of studies lend support to the modern theories of working memory, which postulate an active system responsible for storage and operations. The results also provide additional validation for the automated operation span test created by Unsworth, Heitz, Schrock, & Engle (2003) such that participants who performed poorly on the automated operation span test also performed poorly using a system that is designed in a manner that would be expected to tax such resources. Further, this study provides evidence that theories of visual search behavior can extend to auditory lists and serves as an initial attempt to apply signal detection theory to a new realm. The experiment also has very important practical implications for all systems with auditory menus (particularly IVRs) as it provides empirical evidence that recommends a design practice that is counter to a currently-assumed design best practice.

APPENDIX A

SPEECH USER INTERFACE COMMANDS

The following is a list of the speech user interface commands for the Lotus Notes email system and the result of speaking each command:

1. **Next** – Plays the next message in the selected list

2. **Previous** – Plays the message that appears prior to the currently selected message

3. **Repeat** – Replays the currently selected message

4. **Delete** – After confirmation, deletes the currently selected message and plays the next message in the list

5. **Reply** – Sends a reply to the sender of the currently selected message. First, provides the user with the opportunity to attach a voice recording.

6. **List recipients** – Plays the names or email addresses of each recipient of the currently selected message

7. **Reply to all** - Sends a reply to all recipients of the currently selected message. First, provides the user with the opportunity to attach a voice recording.

8. **Forward** – Forwards the message to one or more persons in the user's address book

9. **Mark unread** – Marks the currently selected message as "unread"

10. **Add sender** – Adds the sender of the currently selected message to the user's address book

11. **Time and date** – Plays the time and date that the message was received

APPENDIX B

MENU ITEM DISTANCE MATRIX

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Next (1)	**.000**	**.00**	**.01**	**.37**	**.50**	**.50**	**.50**	**.44**	**.42**	**.50**	**.46**
Previous (2)	**.00**	**.00**	**.01**	**.37**	**.50**	**.50**	**.50**	**.44**	**.42**	**.50**	**.46**
Repeat (3)	**.01**	**.01**	**.00**	**.35**	**.44**	**.48**	**.44**	**.37**	**.44**	**.50**	**.44**
Delete (4)	**.37**	**.37**	**.35**	**.00**	**.46**	**.48**	**.27**	**.25**	**.38**	**.46**	**.48**
Reply (5)	**.50**	**.50**	**.44**	**.27**	**.00**	**.46**	**.00**	**.12**	**.44**	**.48**	**.48**
List Recipients (6)	**.50**	**.50**	**.48**	**.48**	**.46**	**.00**	**.46**	**.46**	**.40**	**.12**	**.27**
Reply to All (7)	**.50**	**.50**	**.44**	**.27**	**.00**	**.46**	**.00**	**.12**	**.44**	**.48**	**.48**
Forward (8)	**.44**	**.44**	**.37**	**.25**	**.12**	**.46**	**.12**	**.00**	**.44**	**.46**	**.50**
Mark Unread (9)	**.42**	**.42**	**.44**	**.38**	**.44**	**.40**	**.44**	**.44**	**.00**	**.40**	**.17**
Add Sender (10)	**.50**	**.50**	**.50**	**.46**	**.48**	**.12**	**.48**	**.46**	**.43**	**.00**	**.35**
Timd and Date (11)	**.46**	**.46**	**.44**	**.48**	**.48**	**.27**	**.48**	**.50**	**.17**	**.35**	**.00**

APPENDIX C

LONG AND SHORT IVR DESIGNS REPRESENTED IN VXML 1.0

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">

<var name="helpcounter" expr="0"/>
<var name="goback" expr="'undefined'"/>
<var name="currentversion" expr="'long'"/>
<var name="currentform" expr="'undefined'"/>
<var name="returntomessage" expr="'undefined'"/>
<var name="skiptomenu" expr="'false'"></var>
<var name="menutype" expr="'undefined'"></var>

<link next="#startover">
 <grammar>(main menu) | (start over)</grammar>
</link>

<link next="#confirmexit">
 <grammar>hang up</grammar>
</link>

<link next="#goback">
```

```
 <grammar>go back</grammar>
</link>

<form id="getpw">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'getpw'"/>
  <assign name="document.goback" expr="'getpw'"/>
 </block>
 <field name='pw'>
  <prompt>
<break msecs="150"/>Hello.  Welcome to e-mail by phone.  What's your password?
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
		<break msecs="150"/>
Black or gold, what's your password?
   </if>
   <if cond="helpcounter == 2">
		<break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  When you started
```

```
this experiment, you should have received a password.  The passwords for this
experiment are black or gold.  What's your password?
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   black | gold
  </grammar>
  <filled>
   <assign name="document.pw" expr="pw"/>
    <if cond="document.pw =='black'">
    <assign name="menutype" expr="'black'"/>
    <goto next="#introlong"/>
    </if>
    <if cond="document.pw =='gold'">
    <assign name="menutype" expr="'gold'"/>
    <goto next="#introshort"/>
    </if>
  </filled>
 </field>
</form>
```

```
<form id="introlong">
<block>
<assign name="document.goback" expr="'introlong'"/>
<break msecs="150"/>Email by phone inbox.
At any time you can say Help, Go Back,
Start Over, or Hang Up.
<goto next="#startlong"/>
</block>
</form>

<form id="introshort">
<block>
<assign name="document.goback" expr="'introshort'"/>
<break msecs="150"/>Email by phone inbox.
At any time you can say Help, Go Back,
Start Over, or Hang Up.
<goto next="#startshort"/>
</block>
</form>

<form id="startlong">
 <block>
```

```
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'startlong'"/>
  <assign name="returntomessage" expr="'startlong'"/>
  <assign name="document.goback" expr="'introlong'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
<break msecs="150"/>Message 1 from Sara Ferguson is about Company Picnic.
<break msecs="250"/>
Hi everyone.  We're all interested in knowing who will be at the picnic this
weekend.  Please respond and let everyone know if you'll be able to make it.
Thanks.  Sara.
            <break msecs="500"/>
 </block>
 <field name='mainlong'>
  <prompt>
<break msecs="150"/>Select Next, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
Reply, <break msecs="750"/> Reply to All, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
```

or Time and Date.

<break msecs="1500"/>

At any time you can say Help, Go Back, Start Over, or Hang Up.

</prompt>

<catch event="help noinput nomatch">

<assign name="helpcounter" expr="helpcounter+1"/>

<if cond="helpcounter == 1">

<break msecs="150"/>

To navigate to the next message, say next.

To repeat this message, say repeat. To delete the message, say delete. To reply to this message say reply. To reply to all recipients, say reply to all. To forward this message, say Forward. To hear who the message was sent to, say List recipients. To mark this message as unread, say mark unread. To add the message sender to your address book, say add sender. To hear the time and date the message was sent, say time and date.

</if>

<if cond="helpcounter == 2">

<break msecs="150"/>

At any time, you can say help, go back, start over, or hang up. For the next message, say next. To repeat this one, say repeat.

To delete it, say delete. To reply only to the message sender, say reply. To reply to all message recipients, say reply to all. To forward the message, say forward.

```
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
sender. To hear the time and date the message was sent, say time and date.
          <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  next | previous | repeat | delete | reply | reply to all |
  forward | [list] recipients {recipients} | mark unread | add sender
  | time and date {timedate} | time {timedate} | date {timedate}
 </grammar>
 <filled>
  <assign name="document.mainlong" expr="mainlong"/>
   <if cond="document.mainlong =='next'">
   <assign name="skiptomenu" expr="'false'"/>
   <goto next="#longmessage2"/>
   </if>

   <if cond="document.mainlong =='previous'">
   <break msecs="150"/>This is your first message.
   <assign name="skiptomenu" expr="'true'"/>
   <goto next="#startlong"/>
```

```
</if>

<if cond="document.mainlong =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#startlong"/>
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.
Moving to next message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage2"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
```

```
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#startlong"/>
</if>

<if cond="document.mainlong =='mark unread'">
<break msecs="150"/>The message has been marked as unread.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#startlong"/>
</if>
```

```
    <if cond="document.mainlong =='add sender'">
    <break msecs="150"/>Sara Ferguson has been added to your address book.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#startlong"/>
    </if>

    <if cond="document.mainlong =='timedate'">
    <break msecs="150"/>This message was received yesterday at three forty three P M.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#startlong"/>
    </if>

  </filled>
 </field>
</form>

<form id="longmessage2">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'longmessage2'"/>
  <assign name="returntomessage" expr="'longmessage2'"/>
  <assign name="document.goback" expr="'startlong'"/>
```

```
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
Message 2 from John Cardo is about, Pictures.
<break msecs="250"/>
Hey, see the attached pictures.
What a great time, huh?  Please forward these on to Carl Green.  I don't have
his email address.  Thanks, John.
            <break msecs="500"/>
 </block>
 <field name='mainlong'>
  <prompt>
<break msecs="150"/>Select Next, <break msecs="750"/> Previous, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
Reply, <break msecs="750"/> Reply to All, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
or Time and Date.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
```

<catch event="help noinput nomatch">

<assign name="helpcounter" expr="helpcounter+1"/>

<if cond="helpcounter == 1">

<break msecs="150"/>

To navigate to the next message, say next. For the previous, say previous. To repeat this message, say repeat. To delete the message, say delete. To reply to this message say reply. To reply to all recipients, say reply to all. To forward this message, say Forward. To hear who the message was sent to, say List recipients. To mark this message as unread, say mark unread. To add the message sender to your address book, say add sender. To hear the time and date the message was sent, say time and date.

</if>

<if cond="helpcounter == 2">

<break msecs="150"/>

At any time, you can say help, go back, start over, or hang up. For the next message, say next. For the previous, say previous. To repeat this one, say repeat. To delete it, say delete. To reply only to the message sender, say reply. To reply to all message recipients, say reply to all. To forward the message, say forward. To hear a list of the message recipients, say list recipients. To mark the message as unread, say mark unread. To add the message sender to your address book, say add sender. To hear the time and date the message was sent, say time and date.

<assign name="helpcounter" expr="0"/>

```
 </if>
</catch>
<grammar>
 next | previous | repeat | delete | reply | reply to all |
 forward | [list] recipients {recipients} | mark unread | add sender
 | time and date {timedate} | time {timedate} | date {timedate}
</grammar>
<filled>
 <assign name="document.mainlong" expr="mainlong"/>
  <if cond="document.mainlong =='next'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#longmessage3"/>
  </if>

  <if cond="document.mainlong =='previous'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#startlong"/>
  </if>

  <if cond="document.mainlong =='repeat'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#longmessage2"/>
```

```
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.  Moving to next
message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage3"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
```

```
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage2"/>
</if>

<if cond="document.mainlong =='mark unread'">
<break msecs="150"/>The message has been marked as unread.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage2"/>
</if>

<if cond="document.mainlong =='add sender'">
<break msecs="150"/>John Cardo has been added to your address book.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage2"/>
</if>
```

```
    <if cond="document.mainlong =='timedate'">
    <break msecs="150"/>This message was received yesterday at eight sixteen A M.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage2"/>
    </if>

  </filled>
 </field>
</form>


<form id="longmessage3">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'longmessage3'"/>
  <assign name="returntomessage" expr="'longmessage3'"/>
  <assign name="document.goback" expr="'longmessage2'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
```

```
Message 3 from Joe Jacobs is about, Hey.

<break msecs="250"/>

Hey, how's everything?  It was good to

meet you this weekend.  If you're ever back in the Bay area, don't hesitate to

give me a call.  Joe

            <break msecs="500"/>

 </block>

 <field name='mainlong'>

  <prompt>

<break msecs="150"/>Select Next, <break msecs="750"/> Previous, <break msecs="750"/>

Repeat, <break msecs="750"/> Delete, <break msecs="750"/>

Reply, <break msecs="750"/>

Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>

Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>

or Time and Date.

            <break msecs="1500"/>

At any time you can say Help, Go Back, Start Over, or Hang Up.

  </prompt>

  <catch event="help noinput nomatch">

   <assign name="helpcounter" expr="helpcounter+1"/>

   <if cond="helpcounter == 1">

             <break msecs="150"/>
```

```
To navigate to the next message, say next. For the previous, say previous.
To repeat this message, say repeat. To delete the message, say delete. To
reply to this message say reply. .
To forward this message, say Forward. To hear who the message was sent to,
say List recipients.  To mark this message as unread, say mark unread. To add
the message sender to your address book, say add sender.  To hear the time and
date the message was sent, say time and date.
  </if>
  <if cond="helpcounter == 2">
          <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  For the next
message, say next. For the previous, say previous. To repeat this one, say repeat.
To delete it, say delete. To reply only to the message sender, say reply. To forward
the message, say forward.
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
sender. To hear the time and date the message was sent, say time and date.
          <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  next | previous | repeat | delete | reply | reply to all |
```

```
 forward | [list] recipients {recipients} | mark unread | add sender
 | time and date {timedate} | time {timedate} | date {timedate}
</grammar>
<filled>
 <assign name="document.mainlong" expr="mainlong"/>
  <if cond="document.mainlong =='next'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#longmessage4"/>
  </if>

  <if cond="document.mainlong =='previous'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#longmessage2"/>
  </if>

  <if cond="document.mainlong =='repeat'">
  <assign name="skiptomenu" expr="'false'"/>
  <goto next="#longmessage3"/>
  </if>

  <if cond="document.mainlong =='delete'">
```

```
<break msecs="150"/>This message will be deleted when you hang up.  Moving to next
message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage4"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
```

```
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>You are the only recipient of this message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage3"/>
</if>

<if cond="document.mainlong =='mark unread'">
<break msecs="150"/>The message has been marked as unread.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage3"/>
</if>

<if cond="document.mainlong =='add sender'">
<break msecs="150"/>Joe Jacobs has been added to your address book.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage3"/>
</if>

<if cond="document.mainlong =='timedate'">
<break msecs="150"/>This message was received yesterday at eight sixteen A M.
```

```
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage3"/>
    </if>

  </filled>
 </field>
</form>


<form id="longmessage4">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'longmessage4'"/>
  <assign name="returntomessage" expr="'longmessage4'"/>
  <assign name="document.goback" expr="'longmessage3'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
Message 4 from Craig Marshall is about Lunch.
<break msecs="250"/>
Hey, do you have lunch plans?
```

```
Let me know if you'll have time to grab a bite.  Craig
          <break msecs="500"/>
 </block>
 <field name='mainlong'>
  <prompt>
Select Next, <break msecs="750"/> Previous, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
Reply, <break msecs="750"/> Reply to All, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
or Time and Date.
          <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
To navigate to the next message, say next. For the previous, say previous.
To repeat this message, say repeat. To delete the message, say delete. To
reply to this message say reply. To reply to all recipients, say reply to all.
To forward this message, say Forward. To hear who the message was sent to,
```

```
say List recipients.  To mark this message as unread, say mark unread. To add
the message sender to your address book, say add sender.  To hear the time and
date the message was sent, say time and date.
   </if>
   <if cond="helpcounter == 2">
             <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  For the next
message, say next. For the previous, say previous. To repeat this one, say repeat.
To delete it, say delete. To reply only to the message sender, say reply. To reply
to all message recipients, say reply to all. To forward the message, say forward.
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
sender. To hear the time and date the message was sent, say time and date.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   next | previous | repeat | delete | reply | reply to all |
   forward | [list] recipients {recipients} | mark unread | add sender
   | time and date {timedate} | time {timedate} | date {timedate}
  </grammar>
  <filled>
```

```
<assign name="document.mainlong" expr="mainlong"/>
<if cond="document.mainlong =='next'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage5"/>
</if>

<if cond="document.mainlong =='previous'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage3"/>
</if>

<if cond="document.mainlong =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage4"/>
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.  Moving to next
message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage5"/>
</if>
```

```
<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
<assign name="skiptomenu" expr="'true'"/>
```

```
  <goto next="#longmessage4"/>
  </if>

  <if cond="document.mainlong =='mark unread'">
  <break msecs="150"/>The message has been marked as unread.
  <assign name="skiptomenu" expr="'true'"/>
  <goto next="#longmessage4"/>
  </if>

  <if cond="document.mainlong =='add sender'">
  <break msecs="150"/>Joe Jacobs has been added to your address book.
  <assign name="skiptomenu" expr="'true'"/>
  <goto next="#longmessage4"/>
  </if>

  <if cond="document.mainlong =='timedate'">
  <break msecs="150"/>This message was received yesterday at eight sixteen A M.
  <assign name="skiptomenu" expr="'true'"/>
  <goto next="#longmessage4"/>
  </if>

</filled>
```

```
 </field>
</form>



<form id="longmessage5">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'longmessage5'"/>
  <assign name="returntomessage" expr="'longmessage5'"/>
  <assign name="document.goback" expr="'longmessage4'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
Message 5 from Ken Jeffries is about, Party Friday.
<break msecs="250"/>
Hey everyone, I'm having
a party at my place.  Stop by anytime after five and bring some meat if you want
to grill out.  I'll supply the beer.  Ken.
            <break msecs="500"/>
 </block>
 <field name='mainlong'>
```

```
  <prompt>
Select Next, <break msecs="750"/> Previous, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
Reply, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
or Time and Date.
          <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
To navigate to the next message, say next. For the previous, say previous.
To repeat this message, say repeat. To delete the message, say delete. To
reply to this message say reply.
To forward this message, say Forward. To hear who the message was sent to,
say List recipients.  To mark this message as unread, say mark unread. To add
the message sender to your address book, say add sender.  To hear the time and
date the message was sent, say time and date.
   </if>
```

```
  <if cond="helpcounter == 2">
           <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  For the next
message, say next. For the previous, say previous. To repeat this one, say repeat.
To delete it, say delete. To reply only to the message sender, say reply.
To forward the message, say forward.
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
sender. To hear the time and date the message was sent, say time and date.
           <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  next | previous | repeat | delete | reply | reply to all |
  forward | [list] recipients {recipients} | mark unread | add sender
  | time and date {timedate} | time {timedate} | date {timedate}
 </grammar>
 <filled>
  <assign name="document.mainlong" expr="mainlong"/>
   <if cond="document.mainlong =='next'">
   <assign name="skiptomenu" expr="'false'"/>
   <goto next="#longmessage6"/>
```

```
</if>

<if cond="document.mainlong =='previous'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage4"/>
</if>

<if cond="document.mainlong =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage5"/>
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.  Moving to next
message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage6"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
```

```
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage5"/>
</if>

<if cond="document.mainlong =='mark unread'">
```

```
    <break msecs="150"/>The message has been marked as unread.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage5"/>
    </if>

    <if cond="document.mainlong =='add sender'">
    <break msecs="150"/>Joe Jacobs has been added to your address book.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage5"/>
    </if>

    <if cond="document.mainlong =='timedate'">
    <break msecs="150"/>This message was received yesterday at three forty three P M.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage5"/>
    </if>

  </filled>
 </field>
</form>
```

```
<form id="longmessage6">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'longmessage6'"/>
  <assign name="returntomessage" expr="'longmessage6'"/>
  <assign name="document.goback" expr="'longmessage5'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
Message 6 from Mark Riverside is about, Great deals on travel.
<break msecs="250"/>
Hi, pay a
one-time fee of twenty three ninety nine by credit card and receive weekly
emails about travel deals for one full year.  Click the link below to begin.
www.bigbigtraveldeals.com <break msecs="250"/> Mark Riverside
            <break msecs="500"/>
 </block>
 <field name='mainlong'>
  <prompt>
Select Next, <break msecs="750"/> Previous, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
```

```
Reply, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
or Time and Date.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
To navigate to the next message, say next. For the previous, say previous.
To repeat this message, say repeat. To delete the message, say delete. To
reply to this message say reply.
To forward this message, say Forward. To hear who the message was sent to,
say List recipients.  To mark this message as unread, say mark unread. To add
the message sender to your address book, say add sender.  To hear the time and
date the message was sent, say time and date.
   </if>
   <if cond="helpcounter == 2">
             <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  For the next
```

```
message, say next. For the previous, say previous. To repeat this one, say repeat.
To delete it, say delete. To reply only to the message sender, say reply.
To forward the message, say forward.
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
sender. To hear the time and date the message was sent, say time and date.
        <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  next | previous | repeat | delete | reply | reply to all |
  forward | [list] recipients {recipients} | mark unread | add sender
  | time and date {timedate} | time {timedate} | date {timedate}
 </grammar>
 <filled>
  <assign name="document.mainlong" expr="mainlong"/>
   <if cond="document.mainlong =='next'">
   <assign name="skiptomenu" expr="'false'"/>
   <goto next="#longmessage7"/>
   </if>

   <if cond="document.mainlong =='previous'">
```

```
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage5"/>
</if>

<if cond="document.mainlong =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage6"/>
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.  Moving to next
message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage7"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>
```

```
<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>You are the only recipient of this message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage6"/>
</if>

<if cond="document.mainlong =='mark unread'">
<break msecs="150"/>The message has been marked as unread.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage6"/>
```

```
    </if>

    <if cond="document.mainlong =='add sender'">
    <break msecs="150"/>Joe Jacobs has been added to your address book.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage6"/>
    </if>

    <if cond="document.mainlong =='timedate'">
    <break msecs="150"/>This message was received yesterday at eight sixteen A M.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage6"/>
    </if>

  </filled>
 </field>
</form>


<form id="longmessage7">
 <block>
  <assign name="helpcounter" expr="0"/>
```

```
  <assign name="currentform" expr="'longmessage7'"/>
  <assign name="returntomessage" expr="'longmessage7'"/>
  <assign name="document.goback" expr="'longmessage6'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainlong"/>
  </if>
   <break msecs="150"/>
Message 7 from Laura Harrington is about, Important Daytes.
<break msecs="250"/>
Find attached a
list of important daytes you should each add to your calendars.  Thanks, Laura
            <break msecs="500"/>
 </block>
 <field name='mainlong'>
  <prompt>
Select Previous, <break msecs="750"/>
Repeat, <break msecs="750"/> Delete, <break msecs="750"/>
Reply, <break msecs="750"/> Reply to All, <break msecs="750"/>
Forward, <break msecs="750"/> List Recipients, <break msecs="750"/>
Mark Unread, <break msecs="750"/> Add Sender, <break msecs="750"/>
or Time and Date.
            <break msecs="1500"/>
```

```
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
To navigate to the the previous message, say previous.
To repeat this message, say repeat. To delete the message, say delete. To
reply to this message say reply. To reply to all recipients, say reply to all.
To forward this message, say Forward. To hear who the message was sent to,
say List recipients.  To mark this message as unread, say mark unread. To add
the message sender to your address book, say add sender.  To hear the time and
date the message was sent, say time and date.
   </if>
   <if cond="helpcounter == 2">
           <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.  For the previous
message, say previous. To repeat this one, say repeat.
To delete it, say delete. To reply only to the message sender, say reply. To reply
to all message recipients, say reply to all. To forward the message, say forward.
To hear a list of the message recipients, say list recipients. To mark the message
as unread, say mark unread. To add the message sender to your address book, say add
```

```
sender. To hear the time and date the message was sent, say time and date.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   next | previous | repeat | delete | reply | reply to all |
   forward | [list] recipients {recipients} | mark unread | add sender
   | time and date {timedate} | time {timedate} | date {timedate}
  </grammar>
  <filled>
   <assign name="document.mainlong" expr="mainlong"/>
    <if cond="document.mainlong =='next'">
    This is your last message.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage7"/>
    </if>

    <if cond="document.mainlong =='previous'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#longmessage6"/>
    </if>
```

```
<if cond="document.mainlong =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage7"/>
</if>

<if cond="document.mainlong =='delete'">
<break msecs="150"/>This message will be deleted when you hang up.  Moving to previous message.
<assign name="skiptomenu" expr="'false'"/>
<goto next="#longmessage6"/>
</if>

<if cond="document.mainlong =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
</if>

<if cond="document.mainlong =='reply to all'">
<break msecs="150"/>Replying to all.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
```

```
</if>

<if cond="document.mainlong =='forward'">
<break msecs="150"/>Forwarding message.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#forwardmessage"/>
</if>

<if cond="document.mainlong =='recipients'">
<break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage7"/>
</if>

<if cond="document.mainlong =='mark unread'">
<break msecs="150"/>The message has been marked as unread.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#longmessage7"/>
</if>

<if cond="document.mainlong =='add sender'">
<break msecs="150"/>Joe Jacobs has been added to your address book.
```

```
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage7"/>
    </if>

    <if cond="document.mainlong =='timedate'">
    <break msecs="150"/>This message was received yesterday at three forty three P M.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#longmessage7"/>
    </if>

  </filled>
 </field>
</form>

<form id="startover">
<block>
<break msecs="150"/>Starting over.
<assign name="skiptomenu" expr="'false'"/>
<if cond="menutype == 'black'">
<goto next="#introlong"/>
</if>
<if cond="menutype == 'gold'">
```

```
<goto next="#introshort"/>
</if>
</block>
</form>

<form id='confirmexit'>
 <field name="exitChoice" type="boolean">
  <prompt>
   <break msecs="150"/>
    Do you want to end this call?
  </prompt>
  <catch event="help noinput nomatch">
  <assign name="helpcounter" expr="helpcounter+1"/>
  <if cond="helpcounter == 1">
    <break msecs="150"/>Please say Yes, No, or Repeat.
  </if>
  <if cond="helpcounter == 2">
    <break msecs="150"/>At any time you can say Help,
    Repeat, Go Back, Start Over, or Exit.
    To end the call, say Yes.  To return to Hello Worlds,
    say No.
   <assign name="helpcounter" expr="0"/>
```

```
  </if>
  </catch>
  <filled>
   <if cond="exitChoice">
           <goto next="#exit"/>
           <else/>
            Returning.
           <goto expr="'#'+document.currentform"/>
   </if>
  </filled>
 </field>
</form>

<form id='exit'>
 <block>
  <break msecs="150"/>
   Thanks for using e-mail by phone.  Goodbye!
  <exit/>
 </block>
</form>
```

```
<form id='goback'>
 <block>
  <assign name="skiptomenu" expr="'false'"/>
  <goto expr="'#'+document.goback"/>
 </block>
</form>

<form id='returntomessage'>
 <block>
  <goto expr="'#'+document.returntomessage"/>
 </block>
</form>

<form id="makereply">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'makereply'"/>
  <assign name="document.goback" expr="returntomessage"/>
 </block>
 <field name='mainlong'>
  <prompt>
<break msecs="150"/>
```

```
When ready, say Start Recording, or say Cancel.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
To reply, you must attach a recording.  When you're ready,
say start recording, or say cancel if you do not want to reply.
   </if>
   <if cond="helpcounter == 2">
             <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To record your reply to this message say start recording, or say cancel
if you do not want to reply.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   start {startrecording} | recording {startrecording}
   | start recording {startrecording} | cancel
  </grammar>
  <filled>
```

```
   <assign name="document.mainlong" expr="mainlong"/>
    <if cond="document.mainlong =='cancel'">
    <break msecs="150"/>Reply cancelled.
    <goto next="#returntomessage"/>
    </if>

    <if cond="document.mainlong =='startrecording'">
    <goto next="#startrecording"/>
    </if>

  </filled>
 </field>
</form>

<form id="forwardmessage">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'forwardmessage'"/>
  <assign name="document.goback" expr="returntomessage"/>
 </block>
 <field name='forward'>
  <prompt>
```

```
<break msecs="150"/>
Say the recipient's full name.
 </prompt>
 <catch event="help noinput nomatch">
  <assign name="helpcounter" expr="helpcounter+1"/>
  <if cond="helpcounter == 1">
        <break msecs="150"/>
Say the person's name who you want to forward this message to or say cancel.
  </if>
  <if cond="helpcounter == 2">
        <break msecs="150"/>
At any time, you can say, "Help", "Go back", Start over" or Hang up."
Who would you like to forward this message to?
<break msecs="1250"/> If you do not want to forward this message, say Cancel.
       <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  Carl Green {carlgreen} | cancel
 </grammar>
 <filled>
  <assign name="document.forward" expr="forward"/>
```

```
    <if cond="document.forward =='cancel'">
    <break msecs="150"/>Canceled forwarding message
    <goto next="#returntomessage"/>
    </if>

    <if cond="document.forward =='carlgreen'">
    <goto next="#confirmrecipient"/>
    </if>

  </filled>
 </field>
</form>

<form id="confirmrecipient">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'confirmrecipient'"/>
  <assign name="document.goback" expr="'forwardmessage'"/>
 </block>
 <field name='recipient'>
  <prompt>
<break msecs="150"/>
```

```
Was that Carl Green?
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
            <break msecs="150"/>
Please say Yes, No, or Repeat.
   </if>
   <if cond="helpcounter == 2">
            <break msecs="150"/>
At any time, you can say, "Help", "Go back", Start over" or Hang up."
If you want to send this message to Carl Green, say Yes; otherwise, say "No"
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   yes | no
  </grammar>
  <filled>
   <assign name="document.recipient" expr="recipient"/>
    <if cond="document.recipient =='yes'">
    <break msecs="150"/>Message forwarded successfully to Carl Green.
```

```
    <goto next="#returntomessage"/>
    </if>

    <if cond="document.recipient =='no'">
    <goto next="#forwardmessage"/>
    </if>

  </filled>
 </field>
</form>

<form id="startrecording">
<record name="message" beep="true" maxtime="10s" finalsilence="1s" dtmfterm="true"
type="audio/wav">
<prompt><break msecs="150"/>At the tone, begin recording.</prompt>
<noinput><break msecs="150"/>I didn't hear anything, please try again.</noinput>
</record>
<block>
<goto next="#messagesent"/>
</block>
</form>
```

```
<form id="messagesent">
<block>
<break msecs="150"/>Message sent successfully.
<goto next="#returntomessage"/>
</block>
</form>

<form id="startshort">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'startshort'"/>
  <assign name="returntomessage" expr="'startshort'"/>
  <assign name="document.goback" expr="'introshort'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
<break msecs="150"/>Message 1 from Sara Ferguson is about Company Picnic.
<break msecs="250"/>
Hi everyone.  We're all interested in knowing who will be at the picnic this
weekend.  Please respond and let everyone know if you'll be able to make it.
Thanks.  Sara.
```

```
          <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
          <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
   </if>
   <if cond="helpcounter == 2">
           <break msecs="150"/>
```

```
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   listen [to messages] {listen} | respond | distribution
   | [message] details {details} | go back {goback}
  </grammar>
  <filled>
   <assign name="document.mainshort" expr="mainshort"/>

    <if cond="document.mainshort =='listen'">
    <goto next="#listenmenu"/>
    </if>

    <if cond="document.mainshort =='respond'">
    <goto next="#respondmenu"/>
```

```
    </if>

    <if cond="document.mainshort =='distribution'">
    <goto next="#distributionmenu"/>
    </if>

    <if cond="document.mainshort =='details'">
    <goto next="#detailsmenu"/>
    </if>

    <if cond="document.mainshort =='goback'">
    <goto next="#introshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage2">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage2'"/>
```

```
  <assign name="returntomessage" expr="'shortmessage2'"/>
  <assign name="document.goback" expr="'startshort'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
Message 2 from John Cardo is about, Pictures.
<break msecs="250"/>
Hey, see the attached pictures.
What a great time, huh?  Please forward these on to Carl Green.  I don't have
his email address.  Thanks, John.
            <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
```

```
  <assign name="helpcounter" expr="helpcounter+1"/>
  <if cond="helpcounter == 1">
        <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
  </if>
  <if cond="helpcounter == 2">
        <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
        <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  listen [to messages] {listen} | respond | distribution
```

```
 | [message] details {details} | go back {goback}
</grammar>
<filled>
 <assign name="document.mainshort" expr="mainshort"/>

  <if cond="document.mainshort =='listen'">
  <goto next="#listenmenu"/>
  </if>

  <if cond="document.mainshort =='respond'">
  <goto next="#respondmenu"/>
  </if>

  <if cond="document.mainshort =='distribution'">
  <goto next="#distributionmenu"/>
  </if>

  <if cond="document.mainshort =='details'">
  <goto next="#detailsmenu"/>
  </if>

  <if cond="document.mainshort =='goback'">
```

```
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage3">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage3'"/>
  <assign name="returntomessage" expr="'shortmessage3'"/>
  <assign name="document.goback" expr="'shortmessage2'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
Message 3 from Joe Jacobs is about, Hey.
<break msecs="250"/>
Hey, how's everything?  It was good to
meet you this weekend.  If you're ever back in the Bay area, don't hesitate to
```

```
give me a call.  Joe
          <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
          <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
   </if>
   <if cond="helpcounter == 2">
```

```
            <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
           <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   listen [to messages] {listen} | respond | distribution
   | [message] details {details} | go back {goback}
  </grammar>
  <filled>
   <assign name="document.mainshort" expr="mainshort"/>

    <if cond="document.mainshort =='listen'">
    <goto next="#listenmenu"/>
    </if>

    <if cond="document.mainshort =='respond'">
```

```
    <goto next="#respondmenu"/>
    </if>

    <if cond="document.mainshort =='distribution'">
    <goto next="#distributionmenu"/>
    </if>

    <if cond="document.mainshort =='details'">
    <goto next="#detailsmenu"/>
    </if>

    <if cond="document.mainshort =='goback'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage4">
 <block>
```

```
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage4'"/>
  <assign name="returntomessage" expr="'shortmessage4'"/>
  <assign name="document.goback" expr="'shortmessage3'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
Message 4 from Craig Marshall is about Lunch.
<break msecs="250"/>
Hey, do you have lunch plans?
Let me know if you'll have time to grab a bite.  Craig
            <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
```

```
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
         <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
   </if>
   <if cond="helpcounter == 2">
         <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
        <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
```

```
 listen [to messages] {listen} | respond | distribution
 | [message] details {details} | go back {goback}
</grammar>
<filled>
 <assign name="document.mainshort" expr="mainshort"/>

  <if cond="document.mainshort =='listen'">
  <goto next="#listenmenu"/>
  </if>

  <if cond="document.mainshort =='respond'">
  <goto next="#respondmenu"/>
  </if>

  <if cond="document.mainshort =='distribution'">
  <goto next="#distributionmenu"/>
  </if>

  <if cond="document.mainshort =='details'">
  <goto next="#detailsmenu"/>
  </if>
```

```
    <if cond="document.mainshort =='goback'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage5">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage5'"/>
  <assign name="returntomessage" expr="'shortmessage5'"/>
  <assign name="document.goback" expr="'shortmessage4'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
Message 5 from Ken Jeffries is about, Party Friday.
<break msecs="250"/>
Hey everyone, I'm having
```

```
a party at my place.  Stop by anytime after five and bring some meat if you want
to grill out.  I'll supply the beer.  Ken.
        <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
        <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
         <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
   </if>
```

```
  <if cond="helpcounter == 2">
          <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
         <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  listen [to messages] {listen} | respond | distribution
  | [message] details {details} | go back {goback}
 </grammar>
 <filled>
  <assign name="document.mainshort" expr="mainshort"/>

   <if cond="document.mainshort =='listen'">
   <goto next="#listenmenu"/>
   </if>
```

```
    <if cond="document.mainshort =='respond'">
    <goto next="#respondmenu"/>
    </if>

    <if cond="document.mainshort =='distribution'">
    <goto next="#distributionmenu"/>
    </if>

    <if cond="document.mainshort =='details'">
    <goto next="#detailsmenu"/>
    </if>

    <if cond="document.mainshort =='goback'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage6">
```

```
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage6'"/>
  <assign name="returntomessage" expr="'shortmessage6'"/>
  <assign name="document.goback" expr="'shortmessage5'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
Message 6 from Mark Riverside is about, Great deals on travel.
<break msecs="250"/>
Hi, pay a
one-time fee of twenty three ninety nine by credit card and receive weekly
emails about travel deals for one full year.  Click the link below to begin.
www.bigbigtraveldeals.com <break msecs="250"/> Mark Riverside
            <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
```

```
          <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
           <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
this message as unread, select Message Details.
   </if>
   <if cond="helpcounter == 2">
           <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
          <assign name="helpcounter" expr="0"/>
```

```
 </if>
</catch>
<grammar>
 listen [to messages] {listen} | respond | distribution
 | [message] details {details} | go back {goback}
</grammar>
<filled>
 <assign name="document.mainshort" expr="mainshort"/>

  <if cond="document.mainshort =='listen'">
  <goto next="#listenmenu"/>
  </if>

  <if cond="document.mainshort =='respond'">
  <goto next="#respondmenu"/>
  </if>

  <if cond="document.mainshort =='distribution'">
  <goto next="#distributionmenu"/>
  </if>

  <if cond="document.mainshort =='details'">
```

```
    <goto next="#detailsmenu"/>
    </if>

    <if cond="document.mainshort =='goback'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>

<form id="shortmessage7">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'shortmessage7'"/>
  <assign name="returntomessage" expr="'shortmessage7'"/>
  <assign name="document.goback" expr="'shortmessage6'"/>
  <if cond="skiptomenu == 'true'">
   <goto nextitem="mainshort"/>
  </if>
   <break msecs="150"/>
```

```
Message 7 from Laura Harrington is about, Important Dates.
<break msecs="250"/>
Find attached a
list of important dates you should each add to your calendars.  Thanks, Laura
            <break msecs="500"/>
 </block>
 <field name='mainshort'>
  <prompt>
<break msecs="150"/>Select Listen to Messages, <break msecs="750"/>
Respond, <break msecs="750"/> Distribution, <break msecs="750"/> or
Message Details.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
For options that allow you to continue listening to messages,
select Listen to Messages.  To respond to or delete this message, select Respond.
To list the message recipients or add the message sender to your address book,
select Distribution.  To hear the time and date the message was sent or to mark
```

```
this message as unread, select Message Details.
  </if>
  <if cond="helpcounter == 2">
         <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To navigate to another message or to repeat this one, select Listen to Messages.
To reply, forward, or delete this message, select Respond.
To hear the message recipients or add the message sender to your address book,
select Distribution.  For details regarding when the message was sent or
to mark the message as unread, select Message Details.
         <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  listen [to messages] {listen} | respond | distribution
  | [message] details {details} | go back {goback}
 </grammar>
 <filled>
  <assign name="document.mainshort" expr="mainshort"/>

   <if cond="document.mainshort =='listen'">
   <goto next="#listenmenu"/>
```

```
    </if>

    <if cond="document.mainshort =='respond'">
    <goto next="#respondmenu"/>
    </if>

    <if cond="document.mainshort =='distribution'">
    <goto next="#distributionmenu"/>
    </if>

    <if cond="document.mainshort =='details'">
    <goto next="#detailsmenu"/>
    </if>

    <if cond="document.mainshort =='goback'">
    <assign name="skiptomenu" expr="'false'"/>
    <goto next="#startshort"/>
    </if>

  </filled>
 </field>
</form>
```

```
<form id="listenmenu">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'listenmenu'"/>
  <assign name="document.goback" expr="returntomessage"/>
  <assign name="skiptomenu" expr="'true'"/>
 </block>
 <field name='listen'>
  <prompt>
<break msecs="150"/>Select Next, Previous, or Repeat.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
To navigate to the next message, say next; for the previous, say previous;
to repeat this message, say repeat.
   </if>
   <if cond="helpcounter == 2">
```

```
            <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
For the next message, say next, for the previous, say previous,
to repeat this one, say repeat.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   next | previous | repeat
  </grammar>
  <filled>
   <assign name="document.listen" expr="listen"/>

    <if cond="document.listen =='next'">
    <assign name="skiptomenu" expr="'false'"/>
      <if cond="returntomessage =='startshort'">
      <goto next="#shortmessage2"/>
      </if>
      <if cond="returntomessage =='shortmessage2'">
      <goto next="#shortmessage3"/>
      </if>
      <if cond="returntomessage =='shortmessage3'">
```

```
  <goto next="#shortmessage4"/>
  </if>
  <if cond="returntomessage =='shortmessage4'">
  <goto next="#shortmessage5"/>
  </if>
  <if cond="returntomessage =='shortmessage5'">
  <goto next="#shortmessage6"/>
  </if>
  <if cond="returntomessage =='shortmessage6'">
  <goto next="#shortmessage7"/>
  </if>
  <if cond="returntomessage =='shortmessage7'">
  <assign name="skiptomenu" expr="'true'"/>
  <break msecs="150"/>This is your last message.
  <goto next="#shortmessage7"/>
  </if>
</if>

<if cond="document.listen =='previous'">
<assign name="skiptomenu" expr="'false'"/>
  <if cond="returntomessage =='startshort'">
  <assign name="skiptomenu" expr="'true'"/>
```

```
  <break msecs="150"/>This is your first message.
  <goto next="#startshort"/>
  </if>
  <if cond="returntomessage =='shortmessage2'">
  <goto next="#startshort"/>
  </if>
  <if cond="returntomessage =='shortmessage3'">
  <goto next="#shortmessage2"/>
  </if>
  <if cond="returntomessage =='shortmessage4'">
  <goto next="#shortmessage3"/>
  </if>
  <if cond="returntomessage =='shortmessage5'">
  <goto next="#shortmessage4"/>
  </if>
  <if cond="returntomessage =='shortmessage6'">
  <goto next="#shortmessage5"/>
  </if>
  <if cond="returntomessage =='shortmessage7'">
  <goto next="#shortmessage6"/>
  </if>
</if>
```

```
<if cond="document.listen =='repeat'">
<assign name="skiptomenu" expr="'false'"/>
  <if cond="returntomessage =='startshort'">
  <goto next="#startshort"/>
  </if>
  <if cond="returntomessage =='shortmessage2'">
  <goto next="#shortmessage2"/>
  </if>
  <if cond="returntomessage =='shortmessage3'">
  <goto next="#shortmessage3"/>
  </if>
  <if cond="returntomessage =='shortmessage4'">
  <goto next="#shortmessage4"/>
  </if>
  <if cond="returntomessage =='shortmessage5'">
  <goto next="#shortmessage5"/>
  </if>
  <if cond="returntomessage =='shortmessage6'">
  <goto next="#shortmessage6"/>
  </if>
  <if cond="returntomessage =='shortmessage7'">
```

```
      <goto next="#shortmessage7"/>
      </if>
    </if>

  </filled>
 </field>
</form>

<form id="respondmenu">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'respondmenu'"/>
  <assign name="document.goback" expr="returntomessage"/>
  <assign name="skiptomenu" expr="'true'"/>
 </block>
 <field name='respond'>
  <prompt>
<break msecs="150"/>Select Delete, <break msecs="750"/> Reply,
<break msecs="750"/> Reply to All, <break msecs="750"/> or Forward.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
```

```
 <catch event="help noinput nomatch">
  <assign name="helpcounter" expr="helpcounter+1"/>
  <if cond="helpcounter == 1">
        <break msecs="150"/>
To delete the message, say delete; to reply to this message say reply;
to reply to all recipients, say reply to all, To forward this message,
say Forward.
  </if>
  <if cond="helpcounter == 2">
        <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To delete this message, say delete; to reply only to the message sender,
say reply; to reply to all message recipients, say reply to all,
to forward the message, say forward.
       <assign name="helpcounter" expr="0"/>
  </if>
 </catch>
 <grammar>
  delete | reply | reply to all {replyall} | forward
 </grammar>
 <filled>
  <assign name="document.respond" expr="respond"/>
```

```
<if cond="document.respond =='delete'">
<assign name="skiptomenu" expr="'false'"/>
  <if cond="returntomessage =='startshort'">
        <break msecs="150"/>This message will be deleted when you hang up.
            Moving to next message.
  <goto next="#shortmessage2"/>
  </if>
  <if cond="returntomessage =='shortmessage2'">
        <break msecs="150"/>This message will be deleted when you hang up.
            Moving to next message.
  <goto next="#shortmessage3"/>
  </if>
  <if cond="returntomessage =='shortmessage3'">
        <break msecs="150"/>This message will be deleted when you hang up.
            Moving to next message.
  <goto next="#shortmessage4"/>
  </if>
  <if cond="returntomessage =='shortmessage4'">
        <break msecs="150"/>This message will be deleted when you hang up.
            Moving to next message.
  <goto next="#shortmessage5"/>
```

```
  </if>
  <if cond="returntomessage =='shortmessage5'">
       <break msecs="150"/>This message will be deleted when you hang up.
           Moving to next message.
  <goto next="#shortmessage6"/>
  </if>
  <if cond="returntomessage =='shortmessage6'">
       <break msecs="150"/>This message will be deleted when you hang up.
           Moving to next message.
  <goto next="#shortmessage7"/>
  </if>
  <if cond="returntomessage =='shortmessage7'">
       <break msecs="150"/>This message will be deleted when you hang up.
           Moving to previous message.
  <goto next="#shortmessage6"/>
  </if>
</if>

<if cond="document.respond =='reply'">
<break msecs="150"/>Creating reply.
<assign name="skiptomenu" expr="'true'"/>
<goto next="#makereply"/>
```

```
    </if>

    <if cond="document.respond =='replyall'">
    <break msecs="150"/>Replying to all.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#makereply"/>
    </if>

    <if cond="document.respond =='forward'">
    <break msecs="150"/>Forwarding message.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#forwardmessage"/>
    </if>

  </filled>
 </field>
</form>

<form id="distributionmenu">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'distributionmenu'"/>
```

```
  <assign name="document.goback" expr="returntomessage"/>
  <assign name="skiptomenu" expr="'true'"/>
 </block>
 <field name='distribution'>
  <prompt>
<break msecs="150"/>Select list recipients, or add sender.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
To hear who the message was sent to, say List recipients.
To add the message sender to your address book, say add sender.
   </if>
   <if cond="helpcounter == 2">
             <break msecs="150"/>
At any time, you can say help, go back, start over, or hang up.
To hear a list of the message recipients, say list recipients,
to add the message sender to your address book, say add sender.
            <assign name="helpcounter" expr="0"/>
```

```
 </if>
</catch>
<grammar>
 [list] recipients {recipients} | add sender
</grammar>
<filled>
 <assign name="document.distribution" expr="distribution"/>

  <if cond="document.distribution =='recipients'">
   <if cond="returntomessage =='startshort'">
    <break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
    <assign name="skiptomenu" expr="'true'"/>
    <goto next="#startshort"/>
   </if>
   <if cond="returntomessage =='shortmessage2'">
    <break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
    <assign name="skiptomenu" expr="'true'"/>
   <goto next="#shortmessage2"/>
   </if>
   <if cond="returntomessage =='shortmessage3'">
    <break msecs="150"/>You are the only recipient of this message.
    <assign name="skiptomenu" expr="'true'"/>
```

```
<goto next="#shortmessage3"/>
</if>
<if cond="returntomessage =='shortmessage4'">
 <break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage4"/>
</if>
<if cond="returntomessage =='shortmessage5'">
 <break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage5"/>
</if>
<if cond="returntomessage =='shortmessage6'">
 <break msecs="150"/>You are the only recipient of this message.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage6"/>
</if>
<if cond="returntomessage =='shortmessage7'">
 <break msecs="150"/>The recipients are Charlie Gibson, Joe Simms, and Karen Reid.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage7"/>
</if>
```

```
</if>

<if cond="document.distribution =='add sender'">
  <if cond="returntomessage =='startshort'">
   <break msecs="150"/>Sara Ferguson has been added to your address book.
   <assign name="skiptomenu" expr="'true'"/>
   <goto next="#startshort"/>
  </if>
  <if cond="returntomessage =='shortmessage2'">
   <break msecs="150"/>John Cardo has been added to your address book.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage2"/>
  </if>
  <if cond="returntomessage =='shortmessage3'">
   <break msecs="150"/>Joe Jacobs has been added to your address book.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage3"/>
  </if>
  <if cond="returntomessage =='shortmessage4'">
   <break msecs="150"/>Craig Marshall has been added to your address book.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage4"/>
```

```
      </if>
      <if cond="returntomessage =='shortmessage5'">
       <break msecs="150"/>Ken Jeffries has been added to your address book.
       <assign name="skiptomenu" expr="'true'"/>
      <goto next="#shortmessage5"/>
      </if>
      <if cond="returntomessage =='shortmessage6'">
       <break msecs="150"/>Mark Riverside has been added to your address book.
       <assign name="skiptomenu" expr="'true'"/>
      <goto next="#shortmessage6"/>
      </if>
      <if cond="returntomessage =='shortmessage7'">
       <break msecs="150"/>Laura Harrington has been added to your address book.
       <assign name="skiptomenu" expr="'true'"/>
      <goto next="#shortmessage7"/>
      </if>
     </if>

   </filled>
  </field>
 </form>
```

```
<form id="detailsmenu">
 <block>
  <assign name="helpcounter" expr="0"/>
  <assign name="currentform" expr="'detailsmenu'"/>
  <assign name="document.goback" expr="returntomessage"/>
  <assign name="skiptomenu" expr="'true'"/>
 </block>
 <field name='details'>
  <prompt>
<break msecs="150"/>Select mark unread or time and date.
            <break msecs="1500"/>
At any time you can say Help, Go Back, Start Over, or Hang Up.
  </prompt>
  <catch event="help noinput nomatch">
   <assign name="helpcounter" expr="helpcounter+1"/>
   <if cond="helpcounter == 1">
             <break msecs="150"/>
To mark this message as unread, say mark unread,
To hear the time and date the message was sent, say time and date.
   </if>
   <if cond="helpcounter == 2">
             <break msecs="150"/>
```

```
At any time, you can say help, go back, start over, or hang up.
To mark the message as unread, say mark unread,
to hear the time and date the message was sent, say time and date.
            <assign name="helpcounter" expr="0"/>
   </if>
  </catch>
  <grammar>
   mark unread | time and date {timedate} | time {timedate} | date {timedate}
  </grammar>
  <filled>
   <assign name="document.details" expr="details"/>

    <if cond="document.details =='mark unread'">
      <if cond="returntomessage =='startshort'">
       <break msecs="150"/>The message has been marked as unread.
       <assign name="skiptomenu" expr="'true'"/>
       <goto next="#startshort"/>
      </if>
      <if cond="returntomessage =='shortmessage2'">
       <break msecs="150"/>The message has been marked as unread.
       <assign name="skiptomenu" expr="'true'"/>
      <goto next="#shortmessage2"/>
```

```
</if>
<if cond="returntomessage =='shortmessage3'">
 <break msecs="150"/>The message has been marked as unread.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage3"/>
</if>
<if cond="returntomessage =='shortmessage4'">
 <break msecs="150"/>The message has been marked as unread.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage4"/>
</if>
<if cond="returntomessage =='shortmessage5'">
 <break msecs="150"/>The message has been marked as unread.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage5"/>
</if>
<if cond="returntomessage =='shortmessage6'">
 <break msecs="150"/>The message has been marked as unread.
 <assign name="skiptomenu" expr="'true'"/>
<goto next="#shortmessage6"/>
</if>
<if cond="returntomessage =='shortmessage7'">
```

```
  <break msecs="150"/>The message has been marked as unread.
  <assign name="skiptomenu" expr="'true'"/>
 <goto next="#shortmessage7"/>
 </if>
</if>

<if cond="document.details =='timedate'">
 <if cond="returntomessage =='startshort'">
  <break msecs="150"/>This message was received yesterday at three forty three P M.
  <assign name="skiptomenu" expr="'true'"/>
  <goto next="#startshort"/>
 </if>
 <if cond="returntomessage =='shortmessage2'">
  <break msecs="150"/>This message was received yesterday at eight sixteen A M.
  <assign name="skiptomenu" expr="'true'"/>
 <goto next="#shortmessage2"/>
 </if>
 <if cond="returntomessage =='shortmessage3'">
  <break msecs="150"/>This message was received yesterday at eight sixteen A M.
  <assign name="skiptomenu" expr="'true'"/>
 <goto next="#shortmessage3"/>
 </if>
```

```
  <if cond="returntomessage =='shortmessage4'">
   <break msecs="150"/>This message was received yesterday at eight sixteen A M,
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage4"/>
  </if>
  <if cond="returntomessage =='shortmessage5'">
   <break msecs="150"/>This message was received yesterday at three forty three P M.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage5"/>
  </if>
  <if cond="returntomessage =='shortmessage6'">
   <break msecs="150"/>This message was received yesterday at eight sixteen A M.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage6"/>
  </if>
  <if cond="returntomessage =='shortmessage7'">
   <break msecs="150"/>This message was received yesterday at three forty three P M.
   <assign name="skiptomenu" expr="'true'"/>
  <goto next="#shortmessage7"/>
  </if>
 </if>
```

```
  </filled>
 </field>
</form>


</vxml>
```

APPENDIX D

POST-STUDY SYSTEM USABILITY QUESTIONNAIRE (PSSUQ)

1. Overall, I am satisfied with how easy it is to use this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

2. It was simple to use this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

3. I was able to complete the tasks and scenarios quickly using this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

4. I felt comfortable using this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

5. It was easy to learn to use this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

6. I believe I could become productive quickly using this system.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

7. The system gave error messages that clearly told me how to fix problems.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

8. Whenever I made a mistake using the system, I could recover easily and quickly.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

9. The information provided with the system was clear.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

10. It was easy to find the information I needed.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

11. The information was effective in helping me complete the tasks and scenarios.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

12. The organization of information on the system screens was clear.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

Note: *The "interface" includes those items that you use to interact with the system; in this case, the voice prompts.*

13. The interface of this system was pleasant.

STRONGLY								**STRONGLY**
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

14. I liked using the interface of this system.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

15. This system has all the functions and capabilities I expect it to have.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

16. Overall, I am satisfied with this system.

STRONGLY								STRONGLY
AGREE	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**DISAGREE**

COMMENTS:

APPENDIX E

IVR TASKS AND TASK ORDERS

Task Order A

1. Craig Marshall sent a note earlier asking if you had lunch plans. You've been waiting to see how long your 11:00 meeting was going to last and now you see that you won't be able to meet Craig for lunch. Access Craig's message and reply to him appropriately.
2. Access the message from Laura Harrington and mark it as an unread message so it will catch your attention when you access the system next from your PC.
3. Find the message from Joe Jacobs and then add him to your address book
4. Your co-worker, Ken Jeffries sent you a message inviting you to a party at his place. You're interested in finding out which of your co-workers he also invited. Find the message and then check to see who else Ken sent the invite to.
5. Check your mail for any messages from Jon Cardo.
    a. What time did he send the note?
    b. Follow through with his request
6. Find and delete the message from Mark Riverside.
7. Find the message with subject "Company Picnic" and reply to the sender and all recipients with the following message: "Count me in. I'll be there. See everyone this weekend"

Task Order B

1. Find and delete the message from Mark Riverside.
2. Your co-worker, Ken Jeffries sent you a message inviting you to a party at his place. You're interested in finding out which of your co-workers he also invited. Find the message and then check to see who else Ken sent the invite to.
3. Find the message with subject "Company Picnic" and reply to the sender and all recipients with the following message: "Count me in. I'll be there. See everyone this weekend"
4. Find the message from Joe Jacobs and then add him to your address book
5. Craig Marshall sent a note earlier asking if you had lunch plans. You've been waiting to see how long your 11:00 meeting was going to last and now you see that you won't be able to meet Craig for lunch. Access Craig's message and reply to him appropriately.
6. Access the message from Laura Harrington and mark it as an unread message so it will catch your attention when you access the system next from your PC.
7. Check your mail for any messages from Jon Cardo.
    a. What time did he send the note?
    b. Follow through with his request

Task Order C

1. Access the message from Laura Harrington and mark it as an unread message so it will catch your attention when you access the system next from your PC.
2. Find the message with subject "Company Picnic" and reply to the sender and all recipients with the following message: "Count me in. I'll be there. See everyone this weekend"
3. Find and delete the message from Mark Riverside.
4. Check your mail for any messages from Jon Cardo.
    a. What time did he send the note?
    b. Follow through with his request
5. Find the message from Joe Jacobs and then add him to your address book
6. Craig Marshall sent a note earlier asking if you had lunch plans. You've been waiting to see how long your 11:00 meeting was going to last and now you see that you won't be able to meet Craig for lunch. Access Craig's message and reply to him appropriately.
7. Your co-worker, Ken Jeffries sent you a message inviting you to a party at his place. You're interested in finding out which of your co-workers he also invited. Find the message and then check to see who else Ken sent the invite to.

Task Order D

1. Find the message from Joe Jacobs and then add him to your address book
2. Find and delete the message from Mark Riverside.
3. Your co-worker, Ken Jeffries sent you a message inviting you to a party at his place. You're interested in finding out which of your co-workers he also invited. Find the message and then check to see who else Ken sent the invite to.
4. Find the message with subject "Company Picnic" and reply to the sender and all recipients with the following message: "Count me in. I'll be there. See everyone this weekend"
5. Craig Marshall sent a note earlier asking if you had lunch plans. You've been waiting to see how long your 11:00 meeting was going to last and now you see that you won't be able to meet Craig for lunch. Access Craig's message and reply to him appropriately.
6. Access the message from Laura Harrington and mark it as an unread message so it will catch your attention when you access the system next from your PC.
7. Check your mail for any messages from Jon Cardo.
    a. What time did he send the note?
    b. Follow through with his request

APPENDIX F

INFORMED CONSENT FORM

I INFORMED CONSENT TO PARTICIPATE

"Research on the Relationship between Working Memory and Interactive Voice Response Performance and Satisfaction"

A research project is being conducted on the Relationship between Working Memory and Interactive Voice Response Performance and Satisfaction by Patrick Commarford at the University of Central Florida. The purpose of the study is to determine how working memory is utilized when attempting to complete phone-based tasks.

You are being asked to take part in this study by completing a computerized working memory capacity test, and, depending on the outcome of the initial test, by completing a set of phone-based tasks with a voice response system, and by providing us with feedback regarding your level of satisfaction with each task and with the voice response system as a whole. The working memory capacity test will require approximately 15 minutes to complete and the phone-based tasks will take 30 – 60 minutes to complete. Participants completing the initial test only will receive one point of credit for participation. Participants completing all tasks and questionnaires will receive three points for their participation. Please be aware that the phone based tasks will be recorded, so that we can review the data at a later time. Please be aware that you are not required to participate in this research and you may discontinue your participation at any time without penalty. You may also omit any items on the questionnaire(s) you prefer not to answer.

There are no risks associated with participation in this study. If you have further questions about your rights, information is available from the contact address listed at the end of this consent form.

Your responses and the audio recordings will be analyzed and reported anonymously to protect your privacy. All paper-based data will be stored in a locked filing cabinet and all electronic data will be stored on a password protected computer system. All research information gathered will be reported in aggregate form only and your name will not be used in the reporting or publications that may result from the data gathered. Potential benefits associated with the study include a greater understanding of the role that working memory plays in interactive voice response performance and satisfaction. If you agree to voluntarily participate in this research project as described, please indicate your agreement by completing and returning the attached questionnaire. Please *retain* this consent cover form for your reference, and thank you for your participation in this research.

If you believe you have been injured during participation in this research project, you may file a claim with UCF Environmental Health & Safety, Risk and Insurance Office, P.O. Box 163500, Orlando, FL 32816-3500 (407) 823-6300. The University of Central Florida is an agency of the State of Florida for purposes of sovereign immunity and the university's and the state's liability for personal injury or property damage is extremely limited under Florida law. Accordingly, the university's and the state's ability to compensate you for any personal injury or property damage suffered during this research project is very limited.

Institutional Review Board (IRB)
University of Central Florida (UCF)
12443 Research Parkway, Suite 304
Orlando, Florida 32826-3252
Telephone: (407) 823-2901

INFORMED CONSENT TO PARTICIPATE

"Research on Relationship between Working Memory and Interactive Voice Response Performance and Satisfaction"

Print Name: ________________________________

I am 18 years of age or older. I have read the “Informed Consent to Participate” and agree to allow Patrick Commarford to use the information I provide to conduct his research.

____

Signature Date

Principle Investigator:

Name: Patrick Commarford
(561) 414-5294
Commarford@yahoo.com

Student Supervisor:

Janan Al-Awar Smither, Ph.D.
(407) 823-5859
smither@mail.ucf.edu

APPENDIX G

BACKGROUND QUESTIONNAIRE

Background Questionnaire

1. Age __________

2. Gender: M   F

3. Do you have any speech or hearing impairments?   Y   N

4. Do you have at least 3 months experience using a corporate (e.g., Microsoft Outlook, Lotus Notes) or web-based (e.g., Yahoo, Hotmail, AOL) email program?

   Y   N

   Which program(s) ____

   For how long? ______________________

5. Do you have experience designing or programming phone-based or voice user interface applications?

   Y   N

APPENDIX H

UCF IRB APPROVAL FORM

Title of Project: “Research on the Relationship between Working Memory and Interactive Voice Response Performance and Satisfaction”

Principle Investigator:

Signature:

Name: Patrick Commarford
Degree: BA
Title: Human Factors and Applied Experimental Psychology Student
Department: Psycholgy
College: Arts and Sciences
Telephone: 561.414.5294
Facsimile: 561.862.2988
Email: Commarford@yahoo.com

Student Supervisor:

Janan Al-Awar Smither, Ph.D.
Associate Professor
Department of Psychology
University of Central Florida
Orlando, FL 32816-1390
(407) 823-5859 Phone
(407) 823-5862 FAX
smither@mail.ucf.edu

Dates of Proposed Project: From August 2005 until completion of data collection

Source of Funding for the Project: NA

Scientific purpose: The study will provide evidence that supports either a traditional view of the relationship between a short-term memory system and IVR use or supporting arguments derived from more modern theories of working memory (e.g., Baddeley & Hitch, 1974). This study will also seek to determine whether arguments derived from the Criterion-Based Decision Model (MacGregor, Lee, & Lam, 1986) of search extend to auditory search behavior and whether Signal Detection Theory payoffs influence IVR users' response criterion. From an applied perspective, the study will provide evidence that has strong implications for menu design, particularly in the realm of speech-enabled telephony systems.

Describe the Research Methodology in Non-Technical Language:
Participants will first complete a working memory capacity test, which measures their ability to maintain and process information. Participants who score in the upper and lower quartiles will continue to the main study. These participants will complete a series of phone-based tasks using

one of two voice email applications. Following each phone-based task, participants will provide task satisfaction ratings. Upon completion of all tasks, participants will complete a questionnaire, which provides overall system satisfaction ratings. Task performance will be measured by total time on task, total number of errors, and successful IVR task completion rate. Satisfaction will be measured via the After Scenario Questionnaire (ASQ) and the Post-Study System Usability Questionnaire (PSSUQ). Additional classification measures (SDT outcome and search strategy) will also be recorded.

Performance and Satisfaction Measures:
Task performance will be measured by total time on task, total number of errors, and successful IVR task completion rate. Satisfaction will be measured via the After Scenario Questionnaire (ASQ) and the Post-Study System Usability Questionnaire (PSSUQ). Additional classification measures (SDT outcome and search strategy) will also be recorded.

Potential Benefits and Anticipated Risks.
There is no risk to the Human subjects. Data collection will have no direct impact on the participant. In no case will social security numbers, names, or any other individual student identification information be used in the publication of this study. All paper-based data will be stored in a locked filing cabinet and all electronic data will be stored on a password protected computer system.

Benefits:
This study will advance our understanding of working memory, working memory capacity, signal detection, auditory search and the relationships amongst these theories and measurements. Further, the study will provide information which can be directly applied to the design of any user interface that incorporates auditory menus, particularly IVR systems. The information gained will allow designers to create more efficient, usable, and satisfying systems.

Describe how participants will be recruited, the age, the number and proposed compensation.

Undergraduate students enrolled in General Psychology (PSY 2012) will be recruited via Experimentrak. All students will receive course credit for participation in the study. All participants must be at least 18 years of age. There are no upper age restrictions.

The main study will require 132 male and female participants. To examine working memory capacity effects, all participants of the main study must have working memory capacity scores that are in the top or bottom quartiles; therefore approximately 264 participants will complete the automatic operation span test. Half of these participants will be excused and half will continue with the main study. All participants will have at least three months experience using corporate or web email.

All participants will be screened for hearing and speech disorders (self-report), as it would be difficult or impossible for these participants to complete IVR-based tasks. Participants will also complete a demographic questionnaire, which will provide information regarding their familiarity with technology and other background information.

In no case will social security numbers, names, or any other individual student identification information be used in the publication of this study.

Describe the informed consent process (copy of form included)
Student consent will be obtained at the beginning of the session using the attached consent form.

No deception is necessary for this study.

Minors will not be included in this data set.

I approve this protocol for submission to the UCFIRB.

____
Student Supervisor
Janan Al-Awar Smither, Ph.D.

____
Department Chair/Director

University of Central Florida

Office of Research & Commercialization

August 18, 2005

Patrick Commarford
292 Coral Trace Ct
Delray Beach, Fl 33445

Dear Mr. Commarford:

With reference to your protocol #05-2773 entitled, **"Research on the Relationship between Working Memory and Interactive Voice Response Performance and Satisfaction"** I am enclosing for your records the approved, expedited document of the UCFIRB Form you had submitted to our office. **This study was approved by the Chairman on 8/15/05. The expiration date for this study will be 8/14/06.** Should there be a need to extend this study, a Continuing Review form must be submitted to the IRB Office for review by the Chairman or full IRB at least one month prior to the expiration date. This is the responsibility of the investigator. **Please notify the IRB when you have completed this study.**

Please be advised that this approval is given for one year. Should there be any addendums or administrative changes to the already approved protocol, they must also be submitted to the Board through use of the Addendum/Modification Request form. Changes should not be initiated until written IRB approval is received. Adverse events should be reported to the IRB as they occur.

Should you have any questions, please do not hesitate to call me at 407-823-2901.

Please accept our best wishes for the success of your endeavors.

Cordially,

Barbara Ward

Barbara Ward, CIM
IRB Coordinator

Copy: IRB file

BW:cc

12443 Research Parkway • Suite 302 • Orlando, FL 32826-3252 • 407-823-3778 • Fax 407-823-3299

REFERENCES

Abelson, R. P. (1995). *Statistics as Principled Argument.* Mahwah, NJ: Lawrence Earlbaum Associates.

Atkinson, R. C. & Shiffrin, R.M. (1968). Human memory: A proposed system and its control processes. In K.W. Spence & J.T. Spence (Eds.), *The Psychology of Learning and Motivation, Vol 2*. New York: Academic Press.

Atkinson, R. C. & Shiffrin, R.M. (1971). The control of short-term memory. *Scientific American, 225,* 82-90.

Baddeley, A. D. (1999). Working memory. In R. J. Sternberg & R. K. Wagner (Eds.), *Readings in Cognitive Psychology,* (pp. 139-149).

Baddeley, A. D. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences,* 4(11), 417-423.

Baddeley, A. D. (2001). Is Working Memory Still Working? *American Psychologist, 56, (11),* 851-864.

Baddeley, A.D. & Hitch, G. (1974). Working memory. In G.H. Bower (Ed.) *Recent Advances in Learning and Motivation, 8*, Academic Press, New York.

Baddeley, A. D., & Lieberman, K. (1980). Spatial working memory. In R. Nickerson, (Ed.) *Attention and Performance VIII*, (pp. 521-539). Hillsdale, N.J: Erlbaum.

Baddeley, A., Logie, R. & Nimmo Smith, I. (1985). Components of fluent reading. *Journal of Memory and Language, 24*, 119-131.

Baddeley, A.D., Thomson, N., & Buchanan, M. (1975). Word length and the structure of short-term memory. *Journal of Verbal Learning and Verbal Behavior, 14,* 575–589.

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

Barret, L. S., Tugade, M. M., & Engle, R. W. (2004). Individual differences in working memory capacity and dual-process theories of the mind. *Psychological Bulletin, 130(4),* 553 – 573.

Bisseret, A. (1981). Application of signal detection theory to decision making in supervisory control. *Ergonomics, 24,* 81-94.

Brainard, R. W., Irby, T. S., Fitts, P. M., & Alluisi, E. (1962). Some variables influencing the rate of gain of information. *Journal of Experimental Psychology, 63,* 105-110.

Cantor, J., & Engle, R. W. (1993). Working-memory capacity as long-term memory activation: An individual-differences approach. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19(5),* 1101 – 1114.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices test. *Psychological Review, 97(3),* 404 – 431.

Chincotta, D., Underwood, G., Abd Ghani, K., Papadopoulou, E., & Wresinski, M. (1999). Memory span for Arabic numerals and digit words: Evidence for a limited-capacity visuo-spatial storage system. *Quarterly Journal of Experimental Psychology, 2A,* 325-351.

Cohen, M. H., Giangola, J. P., & Balogh, J. (2004). *Voice User Interface Design*: Boston: Addison-Wesley.

Colle, H. A., & Welsh, A. (1976). Acoustic masking in primary memory. *Journal of Verbal Learning and Verbal Behavior, 15*, 17-32

Conrad, R. & Hull, A. J. (1964). Information, acoustic confusion, and memory span. *British Journal of Psychology, 55*, 429-432.

Cook, T. D., & Campbell, D.T. (1979). *Quasi-Experimentation: Design & analysis issues for field settings.* Boston: Houghton Mifflin Co.

Craik, F. I., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning & Verbal Behaviour, 11,* 671-684.

Dalton, P., & Lavie, N. (2004). Auditory attentional capture: Effects of singleton distractor sounds. *Journal of Experimental Psychology: Human Perception & Performance, 30(1),* 180-193.

Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior, 19,* 450-466.

Davis, J. (1967). Auditory search for syllables embedded within meaningful sentences. *Journal of the Acoustical Society of America, 41(5),* 1277-1282.

Dillon, William, R. & Goldstein, M. (1984). *Multivariate Analysis: Methods and Applications.* New York: John Wiley & Sons, Inc.

Eberts, R. E. (1994). *User interface design.* New Jersey: Prentice-Hall, Inc.

Engle, R. W., Cantor, J., & Carullo, J. J. (1992). Individual differences in working memory and comprehension: A test of four hypotheses. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18(5),* 972 – 992.

Engle, R. W., Nations, J. K., & Cantor, J. (1990). Is "working memory capacity" just another name for word knowledge? *Journal of Educational Psychology, 82(4),* 799-804.

Engle, R. W., Tuholski, S. W., Laughlin, J. E., & Conway, A. R. (1999). Working memory, short-term memory, and general fluid intelligence: A latent variable approach. *Journal of Experimental Psychology: General, 128(3),* 309-331.

Eysenck, M. W. (1978). Levels of processing: A critique. *British Journal of Psychology, 69,* 157 – 169.

Frederiksen, J. R. (1969). Response perseveration in auditory word recognition. *Journal of Experimental Psychology, 79(1),* 48-55.

Fisk, A. D., & Rogers, W. A. (1997). *Handbook of Human Factors and the Older Adult.* San Diego: Academic Press.

Gardner, R. C. (2001). *Psychological Statistics using SPSS for Windows.* New Jersey: Prentice Hall.

Gardner-Bonneau, D. J., (1992a). Human factors problems in interactive voice response (IVR) applications: Do we need a guideline/standard? *Proceedings of the Human Factors Society 36th Annual Meeting,* 222-226. Santa Monica, CA: Human Factors Society.

Gardner-Bonneau, D. J., (1992b). Human factors in interactive voice response applications: "Common sense" is an uncommon commodity. *Proceedings of the Human Factors Society 36th Annual Meeting,* 222-226. Santa Monica, CA: Human Factors Society.

Gardner-Bonneau, D. (1999). Guidelines for speech-enabled IVR application design. In D. Gardner-Bonneau (Ed.), *Human Factors and Voice Interactive Systems* (pp. 147-162). Boston: Kluwer Academic Publishers.

Giroux, L., & Belleau, R. (1986). What's on the menu? The influence of menu content on the selection process. *Behaviour and Information Technology, 5,* 169 – 172.

Gould, J. D., Boies, Levy, S. J., Levy, S., Richards, J. T. & Schoonard, J. (1987). The 1984 Olympics message system: A test of behavioral principles of system design. *Communications of the ACM, 30,* 758-769.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory in psychophysics.* New York: Wiley.

Harris, D. H., & Chaney, F. D. (1969). *Human factors in quality assurance.* New York: Wiley.

Hayes, J. M. (1952). Memory span for several vocabularies as a function of vocabulary size. *Quarterly Progress Report, Jan-June,* Massachusetts Institute of Technology: Cambridge.

Harper, M. E., Rhoedenizer-VanDuyne, L, Jentsch, F., Smith-Jentsch, K. A., & Sanchez, A. (2002). Computerized card sort training tool: Is it comparable to manual card sorting? *Proceedings of the Human Factors and Ergonomics Sociecty 46th Annual Meeting*, 2049-2053. Santa Monica, CA: Human Factors and Ergonomics Society.

Hart. S. G., & Staveland, L. E. (1988). Development of NASA TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139-183). Amsterdam, The Netherlands: Elsevier.

Huguenard, B. R., Lerch, F. J., Junker, B.W., Patz, R.J., & Kass, R.E. (1997). Working-memory failure in phone-based interaction. *ACM Transaction on Human-Computer Interaction, 4(2),* 67-102.

Hulme, C, Roodenrys, S, Brown, G, & Mercer R (1995). The role of long-term memory mechanisms in memory span. *British Journal of Psychology, 86* 527-536.

International Business Machines, Corp. (2001). Designing a speech user interface. In *IBM VoiceXML Programmer's Guide*. Author.

Jones, D. (1993). Objects, streams, and threads of auditory attention. In A. D. Baddeley & L. Weiskrantz, (Eds.), *Attention, selection, awareness, and control,* (pp. 87-104). Oxford, England: Oxford University Press.

Just, M. A., & Carpenter, P. A., (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review, 99(1),* 122 – 149.

Kane, M. J., Bleckley, M. K., Conway, A. R., & Engle, R. W. (2001). A controlled-attention view of working memory capacity: Individual differences in memory span and the control of visual orienting. *Journal of Experimental Psychology: General, 130,* 169-183.

Kane, M. J., Hambrick, D. Z., Tulholski, S. W., Wilhelm, O., Payne, T. W., & Engle, R. W. (2004). The generality of working memory capacity: A latent-variable approach to verbal and visuospatial memory span and reasoning. *Journal of Experimental Psychology: General, 133(2),* 189 – 217.

Keller, T. A., Cowan, N., & Saults, J. S. (1995). Can auditory memory for tone pitch be rehearsed? *Journal of Experimental Psychology: Learning, Memory, & Cognition, 21,* 635-645.

Lee, M. D. (2001). Multichannel auditory search: Toward understanding control processes in polychotic auditory listening. *Human Factors, 43(2),* 328-342.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction, 7,* 57-78.

Lewis, J. R. (2002). Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction, 14,* 463-488.

Lockhart, R. S. & Craik, F. I. (1978). Levels of processing: A reply to Eysenck. *British Journal of Psychology, 69,* 171-175.

Logie, R. H., (1986). Visuo-spatial processing in working memory. *Quarterly Journal of Experimental Pyschology, 38A,* 229-247.

Loshe, G., Walker, N., Biolsi, Kl, & Rueter, H. (1991). Classifying graphical information. *Behaviour and Information Technology, 10(5),* 419-436.

MacGregor, J., Lee, E., & Lam, N. (1986). Optimizing the structure of database menu indexes: A decision model of menu search. *Human Factors, 28(4),* 387-399.

Macmillan, N. A. & Creelman, C. D. (2005). *Detection Theory: A User's Guide (2nd Ed.).* Mahwah, NJ: Lawrence Earlbaum Associates.

Marics, M. A., & Engelbeck, G. (1997). Designing voice menu applications for telephones. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 1085-1102). Amsterdam, North Holland: Elsevier.

McGuinness, D. (1983). Sex differences in visual and phonetic search. *Journal of Mental Imagery, Vol 7(1),* 95-104.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review, 63,* 81-97.

Murray, D.J. (1968). Articulation and acoustic confusability in short-term memory. *Journal of Experimental Psychology, 78,* 679–684.

Nelson, T. O. (1977). Repitition and depth of processing. *Journal of Verbal Learning & Verbal Behaviour,16,* 151 – 171.

Paap, K. & Cooke, N. (1997) Design of menus. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 533-572). Amsterdam, North Holland: Elsevier.

Parasuraman, R. & Davies, D. R. (1976). Decision theory of response latencies in vigilance. *Journal of Experimental Psychology, 2(4), 578 – 590.*

Parasuraman, R., Masalonis, A. J., & Hancock, P. A. (2000). Fuzzy signal detection theory: Basic postulates and formulas for analyzing human and machine performance. *Human Factors, 42,* 636-659.

Pierce, B. J., Sisson, N., & Parkinson, S. R. (1992). Menu search and selection processes: a quantitative performance model. *International Journal of Man-Machine Studies, 37,* 679-702.

Pierce, B. J., Parkinson, S. R., & Sisson, N. (1992). Effects of semantic similarity, omission probability and number of alternatives in computer menu search. *International Journal of Man-Machine Studies, 37,* 653-677.

Pike, R., Dalgleish, L., & Wright, J. (1977). A multiple- observations model for response latency and the latencies of correct and incorrect responses in recognition memory. *Memory & Cognition, 5,* 580-589.

Pollack, I. (1953). The assimilation of sequentially encoded information. *American Journal of Psychology, 66,* 421-435.

Prabhakaran, V., Narayanan, K., Zhao, Z., & Gabrelli, J. D. (2000). Integration of diversion information nin working memory in the frontal lobe. *Nature Neuroscience, 3,* 85-90.

Quinn, G., & McConnell, J. (1996). Irrelevant pictures in visual working memory. *Quarterly Journal of Experimental Psychology, 49A,* 200-215.

Redish, G. (2005). How card sorting changed a website team's view of how the sit should be organized. In C. Courage and K. Baxter's *A Practical Guide to User Requirements: Methods, Tools, & Techniques,* (pp. 448-456). San Francisco: Elsevier.

Reid, G. B., & Nygren, T. E. (1988). The subjective workload assessment technique: A scaling procedure for measuring mental workload. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 185-213). Amsterdam, North Holland: Elsevier.

Reisberg, D. (1997). *Cognition: Exploring the science of the mind.* New York: W. W. Norton & Company.

Salame, P., Baddeley, A. (1987). Noise, unattended speech and short-term memory. *Ergonomics, 30,* 1185-1194.

Schumacher, R. M., Jr., Hardzinski, M. L., & Schwartz, A. L. (1995). Increasing the usability of interactive voice response systems: Research and guidelines for phone-based interfaces. *Human Factors, 37(2),* 251-264.

Schwartz, A. L., & Hardzinski, M. L. (1993). *Ameritech phone-based user interface standards and guidelines, 1.0.* Hoffman Estates, IL: Ameritech.

Shah, P., & Miyake, A. (1996). The separability of working memory resources for spatial thinking and language processing: An individual differences approach. *Journal of Experimental Psychology: General, 125,* 4–27.

Smyth, M. M., Pearson, N. A., & Pendleton, L. R. (1988). Movement and working memory: Patterns and positions in space. *Quarterly Journal of Experimental Psychology*, 40, 497-514.

Snowberry, K. Parkinson, S., & Sisson (1983). Computer display menus. *Ergonomics, 26(7),* 699-712.

Turner, M. L. & Engle, R. W. (1989). Is working memory capacity task dependent? *Journal of Memory and Language, 28,* 127-154.

Unsworth, N., Schrock, J. C., Heitz, R. P., & Engle, R. W. (2003). *An automatic version of Ospan.* Poster presented at the annual meeting of the American Psychological Society, Atlanta, GA.

Vanhoucke, V., Neeley, W.L., Mortati, M., Sloan, M., & Nass, C. (2001). Effects of prompt style when navigation through structured data. In *Proceedings of the 8th International Conference on Human-Computer Interaction,* 530-536. Washington: IOS Press.

Vidulich, M. A., & Tsang, P. S. (1986). Techniques of subjective workload assessment: A comparison of SWAT and NASA-bipolar methods. *Ergonomics, 29,* 1385-1398.

Virzi, R. A., & Huitema, J. S. (1997). Telephone-based menus: Evidence that broader is better than deeper. *Proceedings of the Human Factors and Ergonomics Sociecty 41st Annual Meeting*, 315-319. Santa Monica, CA: Human Factors and Ergonomics Society.

Voice Messaging User Interface Forum (1990). Specification document. Cedar Knolls, NJ: Probe Research.

Watkins, M. J. (1977). The intricacy of memory span. *Memory and Cognition, 5(5),* 529-534.

Waugh, N. C., & Norman, D. A. (1965). Primary memory. *Psychological Review, 72(2),* 89-104.

Wickens, C. D. & Hollands, J. G. (2000). *Engineering psychology and human performance* (3rd ed.). New Jersey: Prentice Hall, Inc.

Wickens, C. D., Sandry, D., & Vidulich, M. (1983). Compatibility and resource competition between modalities of input, central processing, and output: Testing a model of complex task performance. *Human Factors, 25,* 227-248.

Wickens, C. D., Vidulich, M., & Sandry-Garza, D. (1984). Principles of S-C-R compatibility with spatial and verbal tasks: The role of display-control location and voice-interactive display-control interfacing. *Human Factors, 26,* 533-543.

Wilkinson, L., & APA Task Force on Statistical Inference. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist, 54,* 594-604.

Zavod, M. J., Rickert, D. E., and Brown, S. H. (2002). The automated card sort as an interface design tool: A comparison of products. *Proceedings of the Human Factors and Ergonomics Sociecty 46th Annual Meeting*, 646-650. Santa Monica, CA: Human Factors and Ergonomics Society.