

# **Advantages and Disadvantages of Press or Say <x> Speech User Interfaces**

Technical Report BCR-UX-2007-0002

November 7, 2007

James R. Lewis

IBM Software Group  
Boca Raton, Florida



## **Abstract**

This report describes the advantages and disadvantages of the common “Press or say <x>” speech user interface. Other than providing a hands-free user interface (which is a characteristic of all speech user interfaces), the press-or-say user interface has all of the limitations of touch-tone without the significant advantages of modern speech recognition.

## **Keywords**

Voice systems

Speech systems

Press or say

Interactive Voice Response (IVR) systems

Speech user interface



# Contents

<b>INTRODUCTION .....</b>	<b>1</b>
<b>THE ADVANTAGE OF AND PROBLEMS WITH “PRESS OR SAY &lt;X&gt;” SPEECH USER INTERFACES.....</b>	<b>1</b>
THE ADVANTAGE OF “PRESS OR SAY <X>” RELATIVE TO TOUCH-TONE .....	1
PROBLEMS WITH “PRESS OR SAY <X>” USER INTERFACES.....	1
<b>WHAT SPEECH USER INTERFACE DESIGNERS SAY ABOUT “PRESS OR SAY &lt;X&gt;” USER INTERFACES.</b>	<b>2</b>
IN BOOKS .....	2
ON THE INTERNET .....	2
<b>WHY ARE “PRESS OR SAY &lt;X&gt;” USER INTERFACES SO PREVALENT? .....</b>	<b>3</b>
EASIEST TRANSITION.....	3
FALSE BELIEF THAT “PRESS OR SAY <X>” IS MORE ROBUST THAN OTHER SPEECH USER INTERFACES.....	3



## **Introduction**

Developers who are converting an application from DTMF (touch-tone) to speech or have prior touch-tone development experience often choose a “Press or say <x>” user interface (“For checking, press or say 1. For savings, press or say 2.”). This report provides information about the advantages and disadvantages of this common speech user interface style.

## **The Advantage of and Problems with “Press or Say <x>” Speech User Interfaces**

### **The Advantage of “Press or Say <x>” Relative to Touch-Tone**

"Press or say <x>" user interfaces are better than plain touch-tone (DTMF) user interfaces because callers can use them without having to move the handset away from their faces to find the right number to press, then having to get the handset back in place to hear the next prompt/message.

### **Problems with “Press or Say <x>” User Interfaces**

The main problem with "Press or say" user interfaces is that they inherit the well-known weaknesses of touch-tone user interfaces and fail to take advantage of the strengths of well-designed speech user interfaces.

One of these well-known weaknesses is that “Press or say” prompts tend to be wordier than better-designed speech prompts – longer to play and thus causing slower interaction. For example, consider, "For checking, press or say 1. For savings, press or say 2.", which is 14 syllables in length. Designers can partially compensate for this by having the system say "press or say" only for the first menu item – "For checking, press or say 1. For savings, 2.", which is 11 syllables long. An appropriate speech prompt is "Select checking or savings", which is 7 syllables shorter than the first version (50% reduction) and 4 syllables shorter than the second version (36% reduction).

Even more important, "Press or say" user interfaces require callers to remember two things about the choices presented in the menu – the content of the choice and the number associated with it. For any caller who has a lower-than-average memory span (limited capability to hold information in short-term memory), this can make it much more difficult to use the system. When callers have a normal short-term memory capacity, these types of user interfaces still require twice as much human memory capacity as a preferred-practice speech menu, in which each options' content is the same as what the caller should say.

# What Speech User Interface Designers Say about “Press or Say <x>” User Interfaces

## In Books

An excellent book on the design of speech user interfaces for telephony applications is "How to Build a Speech Recognition Application: A Style Guide for Telephony Dialogues", by Bruce Balentine and David Morgan (<http://www.amazon.com/How-Build-Speech-Recognition-Application/dp/0967127823>) – it explicitly addresses the use of "Press or say" user interfaces in Section 7.2.1.1 (p. 190):

### 7.2.1.1 Avoid "Press or Say" if Possible

Asking the user to speak a digit for menu selections or other non-numeric data -- simply to emulate the DTMF keypad -- is extremely awkward. Although speech recognition technologies of several years ago were limited to such vocabularies, this is no longer the case.

Another important book on speech user interface design is "Voice User Interface Design", by Michael Cohen, James Giangola, and Jennifer Balogh (<http://www.amazon.com/Voice-Interface-Design-Michael-Cohen/dp/0321185765>), who were voice user interface designers at Nuance at the time they published the book. It does not specifically address "Press or say" user interfaces, but none of the examples presented in the book use "Press or say", which is implicit evidence against the practice.

## On the Internet

In a recent article at telephonyworld.com (<http://www.telephonyworld.com/cgi-bin/news/viewnews.cgi?category=all&id=1150480790>), the speech user interface design experts stated, "Fortunately, we are well beyond the 'Press or say 1' Dark Ages of speech technology," which is very much the belief and practice of professional speech user interface designers.

A Google search (in December, 2006) of "'press or say' speech user interface" turned up these quotes:

"Simple "press or say one" phone trees are rapidly heading for the scrap heap" (<http://www.technologyreview.com/InfoTech/13212/>)

"Presumably this was the method of least effort to add speech to an existing touch-tone dialog, or did some bright spark somewhere think this was a good way to economize on prompts?" (<http://blogs.msdn.com/spokenword/>)

"... excessive demands to remember mappings such as 'press or say one for sports scores,' familiar to most users of voice-based menu systems." (<http://www3.cc.gatech.edu/fce/savoir/pubs/savoir.html>)

"Using your current menus but requiring speech rather than keypad input, or even worse, taking the 'press or say' approach, is a sure way to frustrate callers and increase call length." ([http://www.vanguard.net/DocLib\\_Docs/ICCM\\_Weekly\\_IVR%20\\_Touch-tone\\_to\\_Speech\\_EC\\_0402.pdf](http://www.vanguard.net/DocLib_Docs/ICCM_Weekly_IVR%20_Touch-tone_to_Speech_EC_0402.pdf))

"In the early days, speech systems replicated touch-tone options with prompts such as "Press or say 1." These early systems were effective, but they did not significantly increase automation. Instead of reducing the number and levels of menus, they actually resulted in longer and more tedious prompts." ([http://findarticles.com/p/articles/mi\\_qa3995/is\\_200405/ai\\_n9376505](http://findarticles.com/p/articles/mi_qa3995/is_200405/ai_n9376505))

And so on – I couldn't find any professional speech user interface design web discussion that supported the use of "Press or say <x>" user interfaces.

## Why Are “Press or Say <x>” User Interfaces So Prevalent?

### **Easiest Transition**

A “Press or say <x>” user interface is the easiest way to transition from an existing touch-tone application to a speech-enabled one. Also, imagine that you're someone who has a lot of touch-tone design experience, and now you have the task of designing a speech application. Unless you've made the effort to investigate leading practices in speech user interface design, then this (a "Press or say <x>" UI) is the way you're going to think about structuring the interface. It has one advantage over plain touch-tone (being able to keep a handset to the ear), but it is far from good practice in speech user interface design.

### **False Belief that “Press or Say <x>” is More Robust than Other Speech User Interfaces**

Also, there is a false belief that a “Press or say <x>” user interface is more robust than other speech user interface styles. Part of this belief is based on the fact that a “Press or say <x>” speech user interface uses smaller grammars than other speech user interface styles and will, as a result, have higher recognition accuracy. It is true that the grammars are smaller, but this doesn't necessarily translate to higher recognition accuracy when contrasted against a well-designed alternative. Part of leading practice in designing a directed dialog is to ensure that each option has enough acoustic information for the application to recognize it accurately and that all options are phonetically distinct – advantages that a simple digits grammar cannot match, even in high-noise settings.

It is true that high-noise environments can cause problems with speech recognition, but it has been our experience that noise is more of a problem with false barge-in, which cuts off prompts before they can play. If you disable barge-in to address this, then callers must listen to the entire prompt before they can provide an input, which causes a serious usability problem in that callers must remember the desired choice until the end of the prompt (and calls will take much more time than when barge-in is enabled). Thus, the belief that moving away from a leading practice user interface will result in a better user experience is incorrect.

If there is evidence of a high-noise background (repeated prompt barge-ins with unrecognizable audio), then it is imperative to quickly switch to a touch-tone version of the application, which will stop false prompt cut-off while allowing callers to interrupt prompts with the desired choice by pressing the appropriate key on the keypad. Again, it has been our experience that when callers make a call from a high-noise environment (e.g., an aircraft during a boarding announcement, a noisy roadway), a properly designed application will degrade quickly to touch-tone only (turning off speech) to avoid false prompt cut-off. This aspect of design is, however, independent of whether the speech user interface does or does not use a “Press or say <x>” style.