

# Setting up Experiments to Test a Multi-Threaded Speech User Interface

Alexander Shyrovkov

Electrical and Computer Engineering Department  
University of New Hampshire

## ABSTRACT

The next generation of Speech User Interfaces (SUI) should support multi-threaded dialogues in order to allow concurrent interaction with multiple devices or programs. The SUI will need to support interruptions and resumptions of individual dialogue threads. This paper describes our work on creating tasks suitable for evaluating different approaches the SUI can take to interrupt the human user. We propose four criteria for the tasks and then discuss four different task setups and how well they met the criteria.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces. - Voice I/O, Prototyping.

**General terms:** Experimentation, Human Factors

**Keywords:** Speech interaction, HCI, multi-threaded dialogs

## INTRODUCTION

We envision that next-generation speech user interfaces (SUI) will enable users to engage in multiple tasks at the same time in a hands-busy and eyes-busy environment, resulting in conversations in which threads of dialogues are interleaved. Such a capability would be useful in a number of ubiquitous computing domains where a subject needs to interact with multiple devices in real-time. For example, it would allow a police officer to receive computer-aided dispatches while querying remote license plate databases [1]. It would also allow a driver in a car to catch up on email and occasionally check upcoming traffic conditions or receive navigation instructions. In our definition, each dialogue thread is about a different topic.

The major hypothesis of our research is that we can model the conventions humans use in multi-threaded dialogues with other humans and develop next-generation SUIs based on this model. In human-human interactions multi-threaded dialogues can be illustrated as in Figure 1. Here, the two speakers initially have one dialogue thread. They negotiate a successful switch to a new thread. Before this thread finishes, they switch back to thread 1. After restoring the context, they continue where they left off. They then switch to thread 3 and then back to thread 1. Next there is an unsuccessful attempt to switch back to thread 2, which is shortly followed by a successful switch. We expect that next-generation SUIs will be able to handle these types of interactions.

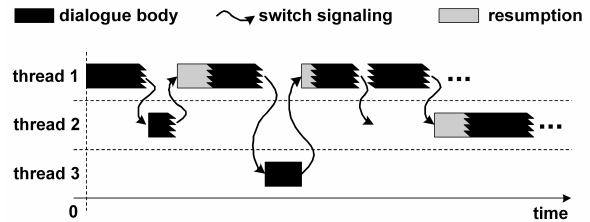


Figure 1: Multi-threaded dialogue.

In human-computer interactions individual dialogue threads will likely be related to specific tasks. For example, in the case of a police officer interacting with a computer, one thread may deal with querying the records of a driver and another thread may deal with finding the location of an accident. In this paper we address the question of computer-initiated interruptions in human-computer interactions. We are interested in finding out how such interruptions influence the performance of the subject on the interrupting task.

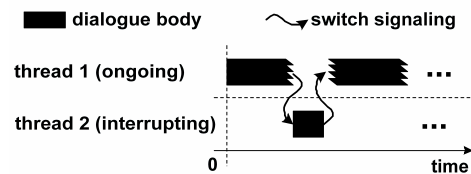


Figure 2: Dialogue threads in human-computer tests.

Our approach is to run experiments in which subjects interact with a computer using a SUI and attempt to complete two tasks at the same time: an ongoing task and an interrupting task (Figure 2). We vary the timing of the interruption and the presence and absence of a beep to signal the interruption. We expect to quantify the effects of the variations in timing and signaling on subject performance on the interrupting task. Since in these experiments we are only interested in the subject's performance on the interrupting task we do not help the subject make the switch back to the ongoing task. The interrupting thread is never itself interrupted.

We believe that in designing next generation SUIs capable of handling multi-threaded dialogues, we must take into account the limits of today's speech recognition technol-

ogy. Hence, we use tasks that are simple enough for today’s speech recognition technology. In this paper, we describe four of the task setups that we have experimented with and the lessons we learned from them.

### BACKGROUND

To model human-human interactions we are running human-human studies [2], in which we do not have to limit our tasks to what can currently be done with speech recognizers. A similar approach was taken by Sidner et al [3] in their experiments in human-robot interaction: they first modeled human-human conventions in activities of interest (e.g. hosting a guest) and then programmed the robot to mimic the human conventions. In using the results from such studies we need to take into account the differences between how people interact with other people and computers [4].

Although there has not been much work done on modeling interruptions in spoken interactions, work has been done on modeling them in several non-spoken types of interactions. Arroyo et al explore using modalities such as heat, smell, sound, vibration and light to signal interruptions [5]. Our investigation of the use of multi-threaded dialogues is similar to cognitive load studies in which subjects switch between two separate manual-visual (that is hands-busy, eyes-busy) tasks [6]. Gillie and Broadbent studied what makes an interruption the most disruptive in the domain of visual tasks [7]. Miyata and Norman give an overview of psychological theory of human behavior when involved in multiple activities and relate it to the design of windows in GUIs [8]. According to Paivio’s dual coding theory [9] humans process visual information and spoken information differently. Therefore, we cannot easily transfer conclusions from experiments with visual tasks into the domain of speech interactions.

### HARDWARE AND SOFTWARE SETUP

Figure 3 shows the hardware setup for our human-computer interaction experiments. We use the AKG HSC 200 SR semi-open headphones with condenser microphone for speech input and output. A USB audio interface (M-Audio MobilePre USB) takes two channels of audio input and records them in a single digital stereo audio file. One channel is used to record the microphone input and the other channel is used to record the computer responses. A notebook PC (PIII-M, 1.13 GHz with 256 MB of RAM) is used to run the software to control the experiment.

Figure 4 shows the software setup. The SUI uses a commercial speech recognition (SR) engine. It also uses a commercial text-to-speech (TTS) engine. Our SUI implements a push-to-talk (PTT) button. Recognition is started when the PTT button is pressed and it is stopped when the button is released. This improves SR performance. SR performance is also improved through the introduction of grammars.

We use XML (Extensible Markup Language) to describe the flow of experiments. The XML file with the test description is interpreted by the XMLT module of the system

(for XML Test). Modifications to a test are done in the XML file. There is no need to recompile the software to change the experiment.

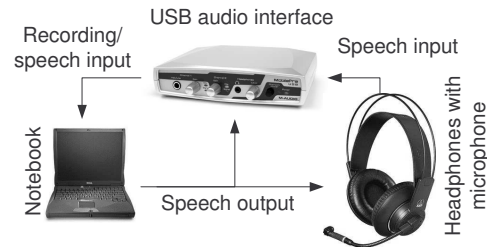


Figure 3: Hardware setup.

During an experiment XMLT generates a log file in XML format that contains the following information: SR results, TTS content, PTT button events (pressing and releasing), information displayed on the screen and expected answers. All entries in the log file have a time stamp that allows synchronization with the digital audio file. The audio file is recorded by a separate process that is built on the Snack Sound Toolkit with Tcl/Tk. The recorder accepts input from the USB audio interface. XMLT sends commands to the recorder to start, stop and save a recording. Communication between XMLT and the recorder is done via TCP/IP. The log file is used to generate an annotation for the digital audio file. This annotation and the audio file are used by DialogueView to visualize the interaction between the subject and the system [10]. This visualization aids in the analysis of the data.

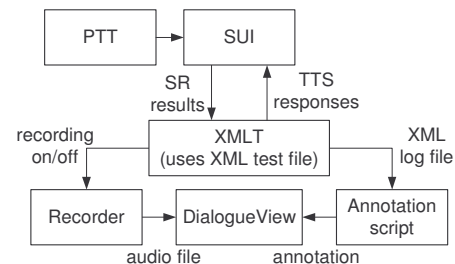


Figure 4: Software setup.

### TASK CRITERIA

In our tests subjects perform two different spoken tasks: an ongoing task and an interrupting task. In designing the experiments we first needed to choose spoken tasks taking into account the following four criteria.

*The performance of subjects on the two tasks when performed separately should be close to error-free. We wanted to design tasks that will be relatively easy for subjects to perform separately. Hence, any subject errors in a multi-threaded dialogue are likely to be caused by the way switching is accomplished between tasks.*

*The performance of subjects on the two tasks when performed in parallel should be worse than when performed separately. We wanted it to be more difficult to perform the tasks in parallel as this would allow us to investigate*

how different ways of interrupting affect the performance on the interrupting task.

*The tasks need to use a limited vocabulary and grammar that can be recognized by a commercial speech recognizer.* We did not want recognition errors to have a major impact on the results.

*The tasks need to be easy to learn for a person.* We wanted to be able to recruit naïve subjects (e.g. undergraduate engineering students), not experts in a given set of activities (e.g. police officers), and still get meaningful results.

#### **FOUR TASK SETUPS**

Using the four criteria we designed four task pairs, described in the following.

##### **Repeating Letters in Sequence and Addition**

In our first setup the ongoing task was to simply say the letters A to F in sequence. There was no confirmation about correctly given letters. In the case of an incorrectly given letter the system would say “Wrong letter” and expect the subject to guess the letter again. The interrupting task was to add two two-digit numbers in base ten. Interruptions would happen after the subject finished speaking. These two tasks satisfied criteria 1, 3 and 4 but did not satisfy criterion 2. We found that it was not very hard for subjects to remember the last letter named, even when an interruption happened. We also found that the interrupting task was very easy and the subjects would make almost no mistakes on it, even when it was performed in parallel with the ongoing task.

##### **Discovering a Sequence of Letters and Addition**

The ongoing task involved the subject guessing a random sequence of the letters A to F (four letters in the sequence). The subject guessed the sequence one letter at a time. For incorrect guesses the system said “Wrong letter” and the subject had to start from the beginning. The interrupting task remained the addition of two two-digit, base ten numbers. Again, the two tasks satisfied criteria 1, 3 and 4 but not 2. The primary task did not require much thinking (it was easy to remember which letter to try next) and the interrupting task posed no problems either, even when the two tasks were performed in parallel.

##### **Rotating a Sequence of Letters and Rotating a Sequence of Numbers**

As the ongoing task the subject had to rotate a sequence of letters. The subject was given a sequence, e.g. ABC and he/she had to rotate this sequence and give the sequences BCA, then CAB and so on. The interrupting task was to rotate a sequence of digits. After implementation of the test we found out that these tasks were either too easy (thus not meeting criterion 2) or too hard (not meeting criterion 4), depending on the length of sequences used.

##### **Addition and Category-Matching Word Detection**

In our fourth setup the ongoing task was to add two four-digit numbers. The interrupting task was to detect one or two words that fit a category, from a sequence of words uttered by the computer. The category was also uttered by

the computer. The interrupting task occurred at random points with respect to the state of the dialogue in the primary task. About 50% of the interruptions were preceded by a beep.

The subject saw the numbers to be added on the computer screen and had to perform the addition without the help of pen and paper. We decided to make the ongoing task addition because in this task the subject needs to remember any carries in partial additions and partial sums. We expected that this would help in meeting criterion 2 (performance worse when tasks performed in parallel), since an interruption would sometimes cause the subject to forget carries or partial sums. We also expected that using base ten for addition would not produce the desired results because we are all trained to do base ten additions on a look-up table basis. Therefore, we made the ongoing task adding two four-digit numbers in bases 6, 7, 8 and 9. Since our student subjects have previously used non-base-10 number systems, this design satisfied criterion 4. We found that on average subjects took about 40 seconds to do the additions without the help of pen and paper.

In the interrupting task the subject was given a sequence of six words. The last word was a category word, which indicated what kind of words should be detected from the sequence of words. For example if the subject was given “yellow, stop, one, stick, three, color”, he/she needed to respond “yellow.” Subjects were asked to detect up to two words from the sequence. This task met criteria 1 through 3 but met criterion 4 only marginally. When subjects were asked to detect a single word in the sequence their performance was acceptable, however detecting two words was difficult.

Since the tasks were close to meeting all four criteria, we conducted initial studies to evaluate the effects of different types of interactions on the subject’s performance on the interrupting task. Our first hypothesis was that interruptions that happened when the subject was speaking would cause more errors in the performance on the interrupting task than interruptions that happened while the subject was thinking. Our second hypothesis was that the presence of the beep would improve performance on the interrupting task.

#### **PRELIMINARY RESULTS**

We tested nine subjects (eight male, one female) on the addition and category-matching word detection task. All were students in the UNH Electrical and Computer Engineering department. Each subject was given training prior to the experiment. The performance of the subjects on the interrupting task alone (control task) was recorded before and after the experiment. The experiment was 20 to 35 minutes long. Interruptions happened at random times about every 40 to 200 seconds and about half were preceded by a beep. Interruptions with and without a beep were randomly ordered.

We processed the parts of the interactions in which the subject had to detect only one word of a certain category.

We looked at 155 such computer initiated interruptions. Of the total, 40 happened while the subject was speaking and 115 when the subject was thinking. Also, 90 were preceded by a beep and 65 were not. We are currently investigating the interactions where the subjects had to detect two words.

During the experiment the subjects made errors 7% of the time when they were interrupted while speaking and 15% of the time when they were interrupted while they were thinking. Since a one-way pair-wise t-test shows that this result has little significance ( $p = 0.112$ ) we cannot draw a definite conclusion about our first hypothesis.

In the control task, subjects made a mistake detecting one word 4% of the time when a beep preceded the task and 32% of the time when it did not. During the experiment the subjects made a mistake 17% of the time with a beep and 6% of the time without the beep. Thus, our second hypothesis that the beep would improve performance was held up during the control task but not when the category matching task was the interrupting task. The difference between the means of error rates (beep vs. no beep) was significant according to the one-way pair-wise t-test with  $p = 0.013$  for the control task and  $p = 0.038$  for the experiment. We do not have an answer to the question why beeps improved performance during the control task and made it worse during the experiment – we hope to find it through further subject studies.

## DISCUSSION

We proposed four criteria that pairs of tasks had to meet to be used in testing multi-threaded SUIs. We designed four pairs of tasks, only one of which met all four criteria. Even in this case criterion 4 was met only marginally (part of the interrupting task was too hard to learn for a person). We feel that part of the problem was that our tasks were boring to work through. In our fourth setup we tried to motivate subjects by telling them they were playing a game, however adding numbers in different bases still proved to be boring. By contrast in our human-human studies [2] we succeeded in making the tasks engaging by creating a scoring mechanism which encouraged the subjects to apply themselves. We feel that the tasks in the human-computer studies described in this paper were boring partly because they were artificially created. The tasks in the human-human studies (playing cards) were a lot closer to realistic tasks. Realistic tasks are also interesting because they are likely to shed light on phenomena relevant to the design of SUIs. Thus, for future research we will make our tasks more engaging and realistic.

We are exploring using more realistic tasks from the domain of law enforcement which we have extensive experience with through our work on the Project54 system [1]. The Project54 system integrates electronic devices in police cruisers and provides a SUI to these devices. We are currently in the process of designing experiments in which a subject would be involved in two tasks related to law-enforcement work. One task would be to perform database queries to find information about license plates. This task

would probably mix visual queues (e.g. license plate numbers displayed on a monitor) and speech interaction with the computer (e.g. dictating the license plate number and listening to query results). Another task would model responding to dispatches from headquarters. This task would primarily require interacting with the computer using speech.

## ACKNOWLEDGEMENTS

This work was funded by the National Science Foundation under IIS-0326496.

## REFERENCES

1. Kun, A.L., Miller, W.T., and Lenharth, W.H. "Computers in police cruisers", *IEEE Pervasive Computing*, 3(4):34-41, 2004.
2. Heeman, P., Yang, F., Kun, A.L, and Shyrovkov, A., "Conventions in Human-Human Multi-Threaded Dialogues: A Preliminary Study", *Intelligent Subject Interfaces Conference*, 2005.
3. Sidner, C., Lee, C., Kidd, C., and Lesh, N., "Explorations in Engagement for Humans and Robots", *IEEE RAS/RSJ International Conference on Humanoid Robots*, 2004
4. Shechtman, N., and Horowitz, L.M., "Media inequality in conversation: How people behave differently when interacting with computers and people", *Human Factors in Computing Systems*, 2003.
5. Arroyo, E., Selker T., and Stouffs, A., "Interruptions as multimodal outputs: Which are the less disruptive?", *International Conference on Multimodal Interfaces*, pp. 479-483, 2002.
6. McFarlane, D.C., "Coordinating the Interruption of People in Human-Computer Interaction", *Human-Computer Interaction*, 1999.
7. Gillie, T., and Broadbent, D., "What makes interruptions disruptive? A study of length, similarity, and complexity", *Psychological Research* 50(4), 1989.
8. Miyata, Y., and Norman, D., "Psychological issues in support of multiple activities", *Subject Centered Systems Design: New Perspectives on Human-Computer Interaction*, 1986.
9. Paivio, A., "Imagery and Verbal Processes", New York: Holt, Rinehart & Winston, 1971.
10. Heeman, P., Yang F., and Strayer, S.E., "DialogueView: An Annotation Tool for Dialogue", *Third SIGdial Workshop on Discourse and Dialogue*, 2002.