What can I say?: evaluating a spoken language interface to email

Marilyn A. Walker, Jeanne Fromer Giuseppe Di Fabbrizio, Craig Mestel, Don Hindle
ATT Labs Research
180 Park Ave., D188
Florham Park, N.J. USA
+1 973 360 8956
{walker, pino,hindle}@research.att.com , jeannie@ai.mit.edu, cmestel@leland.stanford.edu

ABSTRACT
This paper presents experimental results comparing two different designs for a spoken language interface to email. We compare a mixed-initiative dialogue style, in which users can flexibly control the dialogue, to a system-initiative dialogue style, in which the system controls the dialogue. Our results show that even though the mixed-initiative system is more efficient, as measured by number of turns, or elapsed time to complete a set of email tasks, users prefer the system-initiative interface. We posit that these preferences arise from the fact that the system initiative interface is easier to learn and more predictable.

Keywords
Spoken Language Interfaces, Initiative, Email Interfaces

INTRODUCTION
Debate about the utility of spoken language interfaces (SLIs) vs. graphical user interfaces (GUIs) has been on-going for many years. It is widely agreed that GUIs provide immediate feedback, reversible operations, and incrementality, that they support rapid scanning and browsing of information, and that they are easy for novices to use because visual information indicates the currently available options. In contrast, SLIs have been criticized for the inherent limitations of a sequential channel (speech) for delivering information, for requiring the user to learn the language the system can understand, for hiding available command options, and for leading to unrealistic expectations as to their capabilities.

Yet changes in both work practice and technology in the last ten years has greatly increased the potential benefits of SLIs. Workers are much more mobile, making remote access to information stored on a desktop computer highly desirable. The telephone is a lightweight and ubiquitous form of access, and automatic speech recognition (ASR) used in SLIs has vastly improved. These changes mean that many interface designers now believe that SLIs for remote access to a range of personal and public (often web-based) information are both practical and desirable.

However, because it has only been recently possible to build real-time SLIs, very little is known about how to design them to mitigate current limitations in component technologies, such as automatic speech recognition (ASR) and text to speech synthesis (TTS), as well as inherent limitations with SLIs [1,2,3,4,7]. The single most important design choice is the selection of dialogue strategies that the SLI uses in conversations with the user. One key factor is whether the dialogue strategy is system-initiative (SI) or mixed-initiative (MI) [7]. These strategies are illustrated in D1 and D2 for ELVIS (EmaiL Voice Interactive System), an SLI for accessing email by phone:

(D1: System Initiative Strategy)
A: Hi, Elvis here. You have 5 new and 0 unread messages in your inbox. Say Read, or Summarize, or say Help for more options.
U: Read.
A: Select by Content or Position?
U: Content.
A: Select by Sender or Subject?
U: Sender
A: Which Sender?
U: Kim.
A: The message from Kim is about Meeting Tomorrow. The meeting tomorrow is at 10:30 in 2D-516.

(D2: Mixed Initiative Strategy)
A: Hi, Elvis here. I've got your mail.
U: Read me my messages from Kim.
A: The message from Kim is about Meeting Tomorrow. The meeting tomorrow is at 10:30 in 2D-516.

The SI strategy prompts users what to say at each stage of the dialogue with directive prompts [2]. Since prompts must be short, the SI strategy typically executes actions by small increments at each turn. The MI strategy assumes that users know what to say, and never volunteers information. The MI SLI is designed so that users can specify a system function and all of its arguments in a single utterance if they desire. While it might appear obvious that users would

prefer the more flexible interaction style of the MI interface, there are several complicating factors. First, MI requires more knowledge on the part of the users about what they can say, while the SI strategy directs the user. Second, because SI users are directed to produce very short utterances, the ASR performance of SI may be much better. If more error recovery and correction dialogues are necessary for MI, users may find this onerous. Too many spoken language understanding errors may also make it difficult for users to acquire a model of the system's behavior. Without knowing the effect of these factors, it is impossible to predict whether the SI strategy or the MI strategy is better.

This paper discusses the design and evaluation of ELVIS, a research prototype SLI that supports voice access to email by phone. We report experimental results from testing users with both an SI and an MI version of ELVIS. Our experimental data consists of 144 dialogues with 48 users, consisting of a total of 6481 turns. Our results show that even though the MI system is more efficient, as measured by either number of turns, or elapsed time to complete a set of email tasks, users prefer the SI interface.

DESIGNING A SPOKEN LANGUAGE INTERFACE FOR EMAIL

In addition to the dialogue strategy design, a second key aspect of SLI design is deciding what options should be available to the user at each point in the dialogue**.** Previous work has demonstrated the utility of Wizard of Oz studies [1,4,8,9], so we began our design process with a Wizard of Oz (WOZ) study where a person played the part of an SLI for accessing email. We collected, recorded and transcribed 15 extended conversations (1200 utterances) with 6 different prototypical users, mobile professionals accessing their email while away from their office.

We then categorized each utterance in these dialogues in terms of its use of key email access functions. Categories were based on the underlying application, as well as on language-based functionality, such as *reference* to messages by their properties, such as the sender or the subject of the message (e.g. *the message from Kim*), or in context (e.g. as *them, it, that*), Table 1 summarizes the functions used most frequently in the WOZ study. This study suggests that the SLI should minimally support: (1) reading the body of a message and the header information; (2) summarization of the contents of an email folder by content-related attributes such as the sender or subject; (3) selection of individual messages by content fields such as the sender or subject; and (4) request for clarifying help, repetition of something that was said, and undoing of previous actions.

Table 1: Email functions used in Wizard of Oz study

EMAIL ACCESS	N
FUNCTION	
Summarization	20
Reference	101
Folder Action	10
Read Message	67
Search for a Message	8
Message Field Access	5
Repeat	4
Clarifications	37
Help	3

Reading the message and header information requires the use of text-to-speech (TTS) since it is impossible to pre-record messages with a human voice. Reading the body of the message also requires filtering the message body for things that are unpronounceable by TTS, and recognizing attachments in the message body.

In the WOZ study, users typically requested summaries on entering a folder and referred to messages by attributes such as sender or subject in order to randomly select messages of interest in that folder. We hypothesized that summarization and selection capabilities could provide a way to scan and browse information in SLIs. In other words, one way to obviate the limitations of a sequential speech channel is to give users the ability to overview the data (with summarization) and then select the subset of items that they are interested in (with reference). Summarization and selection of messages by content attributes required reproducing searching and sorting functionality available in many email GUI interfaces. For folder summaries, the list of messages had to be converted into a coherent summary that was appropriate to the context, i.e. whether the folder had been created by selecting by sender or by subject.

Even though our WOZ subjects did not typically attempt to access to messages in the sequence in which they were received (or reverse chronological order), we felt that it was necessary to provide this as an additional option because other voice and touch-tone interfaces to voice and email messages provide this option [2,3,9]. Thus both the SI and the MI system support access to messages by relative position within a folder: users can select messages by saying *First*, *Next*, *Previous* and *Last*.

The WOZ study also confirms that the system should provide help so that users can learn its capabilities. We were particularly concerned about the design of help messages for the MI system. The SLI platform that we used to build both versions of ELVIS included a facility for specifying help messages associated with each state of the dialogue. These context-sensitive help messages indicate to the user what command options are available at each point

in the dialogue, and provide a verbal analog to the visible icons in GUIs that indicate the available command options.

Context-sensitive help was available to the user in two ways: at the user's initiative if the user says *Help*, and at the system's initiative with **timeout messages**. The system plays timeout messages when the user doesn't say anything, i.e. after some expected response delay has timed out. The system keeps track of how many times a timeout occurs in each state, so that timeout messages can be modified to be more informative after each timeout.

Because help messages (and email messages) can be long, users must be able to interrupt the system to take control of the interaction at any point, while the system is talking or carrying out a command. This is called **Barge-In.** Supporting barge-in requires that a speech recognizer is always listening, even when the system is currently busy recognizing something the user said previously. Barge-In also involves the ability to abort common procedures in midstream, e.g the system needs to be able to send TTS instructions to stop talking in midstream.

Finally, the SLI must provide some way of undoing a previous command. This is useful in two cases: (1) if the user simply decides they would rather do something different; and (2) if the SLI misunderstand the user. ELVIS supports reversibility by providing an always available *cancel* command that returns the user to the dialogue state before the previous interaction.

EXPERIMENTAL DESIGN

The experiment required users, randomly assigned to either the MI or the SI version of ELVIS, to complete three tasks involving telephone access to email. All of the users regularly used computers in the course of their everyday work and were familiar with email. In one study, the 12 users were administrative assistants or researchers whose area of research was not related to SLIs. We reported results from this study elsewhere [5]. Subsequently, we noticed that response delay was longer than we wanted and that there was a different way of communicating with the email application layer that would significantly reduce it. After implementing the improved version, we then tested another 36 users in both versions of ELVIS. These subjects were summer interns, with little exposure to SLIs, many of whom were not native speakers of English. Below we discuss results from these 36 users.

Experimental instructions were given on three web pages, one for each experimental task. Each web page consisted of a brief general description of Elvis, a list of hints for using Elvis, a task description, and information on calling ELVIS. Subjects read the instructions in their offices before calling ELVIS from their office phone.

Each user performed three tasks in sequence, and each task consisted of two subtasks. Thus the results consisted of 108 dialogues representing 216 attempted subtasks. The task scenarios that the subjects were given were as follows, where subtasks 1.1 and 1.2 were done in the same conversation, similarly for 2.1 and 2.2, and 3.1 and 3.2.

- 1.1: You are working at home in the morning and plan to go directly to a meeting when you go into work. Kim said she would send you a message telling you where and when the meeting is. Find out the Meeting Time and the Meeting Place.
- 1.2: The second task involves finding information in a different message. Yesterday evening, you had told Lee you might want to call him this morning. Lee said he would send you a message telling you where to reach him. Find out Lee's Phone Number.
- 2.1: When you got into work, you went directly to a meeting. Since some people were late, you've decided to call Elvis to check your mail to see what other meetings may have been scheduled. Find out the day, place and time of any scheduled meetings.
- 2.2: The second task involves finding information in a different message. Find out if you need to call anyone. If so, find out the number to call.
- 3.1: You are expecting a message telling you when the Discourse Discussion Group can meet. Find out the place and time of the meeting.
- 3.2: The second task involves finding information in a different message. Your secretary has taken a phone call for you and left you a message. Find out who called and where you can reach them.

These tasks were based on representative tasks from the WOZ study, involving the use of summarization and reference as in Table 1. Each subtask specified the information about criteria for selecting messages, and information within the message body, that the user and the system had to exchange. For example, in scenario 1.1, the user is expecting email from Kim about a meeting and needs to find out the time and place of that meeting (as in Dialogue D1 and D2). Following [6], this scenario is represented in terms of the attribute value matrix (AVM) in Table 2. The AVM representation for all six subtasks is similar to Table 2. Note that the task's information exchange requirement represented in the AVM is independent of the dialogue strategy used to accomplish the task. The use of the AVM to calculate task success is discussed below.

We designed the experimental email folders so that for each task, the desired messages were not among the first two messages (as ordered chronologically). Thus users who accessed messages by chronological order would often have to listen to all five messages in order to complete the task, while users who accessed messages using selection by content could complete the task by listening to two messages. Thus accessing messages by relative position should have led to inefficient dialogues, while the

instructions specified that users should be as efficient as possible and avoid listening to messages unnecessarily.

Table 2: Attribute Value Matrix: Email Scenario Key for Dialogues D1 and D2

ATTRIBUTE	VALUE
Selection Criteria	Kim or Meeting
Email.att1	10:30
Email.att2	2D 516

The general description and the hints on the web page for each task were identical. The subjects were asked to impersonate a different user for each task and were told that they needed to talk to ELVIS to find out some information that had been sent to them in an email message. We decided not to include any specific examples of what users could say in the hints for using ELVIS for three reasons: (1) we wanted the instructions to be identical for both SI and MI; (2) users could get information as to what they could say from the context-sensitive help messages; (3) we wanted to be able to quantify the frequency with which users accessed information on what they could say, and would not have been able to do so if this information had been presented visually. The hints were:

- Anytime you need help with what to say or with what Elvis is doing, you can say *Help*.
- If Elvis misunderstands you and does the wrong thing, you can undo it by saying *Cancel.*
- If you wait too long to tell Elvis what to do, Elvis will tell you what you can do.
- When you are finished with a task, you can go back to the previous context by saying *I'm done here*.
- You don't have to wait for Elvis to finish talking if you've heard enough or you know what you want to do; you can interrupt at any time.

We collected four types of data and extracted a number of variables. First, all dialogues were recorded. The recording supports utterance transcription and measuring aspects of the timing of the interaction, such as whether there were long system response delays, and whether users barged-in on system utterances (the variable named **BargeIn**). BargeIn may reflect learning; as users learn what they can say, they can barge in over the system's utterances. In addition, the recording was used to calculate the total time of the interaction (the variable named **Elapsed Time**).

Second, the system logged its dialogue behavior on the basis of entering and exiting each state in the state transition table for the dialogue. For each state, the system logged the number of timeout prompts (**Timeout Prompts),** the number of times the confidence level for ASR was too low and the system played a special rejection messages, e.g. *Sorry, I didn't understand you* (**ASR Rejections)**, and the times the user said *Help* (**Help Requests**). The number of **System Turns** and the number of **User Turns** were calculated on the basis of this data. In addition, the results of ASR for the user's utterance was logged. A measure of the system's understanding (concept accuracy) was calculated from the recordings in combination with the logged ASR result for each utterance. Mean concept accuracy was then calculated over the whole dialogue to provide a **Mean Recognition Score** (MRS) for the dialogue.

Third, users were required to fill out the web page forms after each task specifying whether they had completed the task and the information they had acquired from the agent (Task Success), e.g. the values for Email.att1 and Email.att2 in Table 2. This supported the use of the Kappa statistic to measure Task Success [6], where Kappa is defined as:

$$K = P(A) – P(E) / 1- P(E)$$

P(A) is the proportion of times that the AVM for the dialogue agrees with the AVM for the scenario key, and P(E) is the proportion of times we would expect the AVMs for the dialogues and keys to agree by chance. When agreement is perfect (all task information items are successfully exchanged), then Kappa=1. When agreement is only at chance, then Kappa=0.

Finally, users responded to a survey on their subjective evaluation of their performance and their satisfaction with the system's performance with the following questions:

- Did you complete the task? (**Comp**)
- Was Elvis easy to understand in this conversation? (**TTS Performance**)
- In this conversation, did Elvis understand what you said? (**ASR Performance**)
- In this conversation, was it easy to find the message you wanted? (**Task Ease**)
- Was the pace of interaction with Elvis appropriate in this conversation? (**Interaction Pace**)
- In this conversation, did you know what you could say at each point of the dialogue? (**User Expertise**)
- How often was Elvis sluggish and slow to reply to you in this conversation? (**System Response**)
- Did Elvis work the way you expected him to in this conversation? (**Expected Behavior**)
- In this conversation, how did Elvis's voice interface compare to the touch-tone interface to voice mail? (**Comparable Interface**)
- From your current experience with using Elvis to get your email, do you think you'd use Elvis regularly to access your mail when you are away from your desk? (**Future Use**).

The user satisfaction survey was multiple choice, and the possible responses to most questions ranged over values such as (*almost never, rarely, sometimes, often, almost always*), or an equivalent range. Each of these responses was mapped to an integer between 1 and 5. Some questions

had (*yes, no, maybe*) responses. Each question emphasized the user's experience with the system in the current conversation, with the hope that satisfaction measures would indicate perceptions specific to each conversation, rather than reflecting an overall evaluation of the system over the three tasks. A Cumulative Satisfaction (CSAT) score for each dialogue was calculated by summing the scores for each question. The survey also included a free text field where users were encouraged to enter any comments they might have.

The goal of the experiment was to evaluate the usability of an SLI for accessing email by phone and to compare the MI dialogue design to the SI dialogue design when the task is held constant. We wished to investigate how users would adapt to the version of the system they were using as they performed a sequence of three similar tasks. Our primary experimental variable was dialogue strategy: whether the user interacted with the SI or the MI version of ELVIS. However, we were also interested in whether the availability of summarization and selection by content increased the functionality of the system. Our hypotheses were:

- H1: The MI strategy is potentially much more efficient than the SI strategy, but its efficiency depends on ASR performance, and the lower the ASR performance the less efficient it will be.
- H2: Users will have trouble knowing what they can say to the MI SLI and this will reduce ASR performance.
- H3: Users' knowledge of what they can say to the MI SLI will improve over the three tasks.
- H4: Because of H1, H2, and H3, Cumulative Satisfaction for the system initiative SLI will be greater for the first task, but Cumulative Satisfaction for the MI SLI will be greater by the third task.
- H5: Use of summarization will increase Cumulative Satisfaction and improve efficiency.
- H6: Use of selection by content will increase Cumulative Satisfaction and improve efficiency.

These hypotheses concern the relation between dialogue strategy, Mean Recognition Score, the utilization of the summarization and selection by content options, and the users' ability to learn what options are available at each point of the dialogue and to acquire a model of the system.

EXPERIMENTAL RESULTS

Our experimental design consisted of two factors; strategy and task. Each of our result measures were analyzed using a two-way ANOVA for these factors. For each result, we report F and p values indicating its statistical significance. Effects that are significant as a function of strategy (SI vs. MI) indicate differences between the two strategies. Effects that are significant as a function of task are potential indicators of learning. We discuss results for each of these factors as they relate to our hypotheses.

We first calculated Task Success in terms of Kappa to see whether task completion rates and scores were affected by dialogue strategy [7]. The average Kappa value over all subjects and tasks was .82, indicating that the task was almost always completed successfully. An ANOVA with Kappa as the dependent variable revealed no significant differences for Kappa as a function of task or strategy.

Hypothesis H1 focuses on the relation between Mean Recognition Score (MRS) and efficiency. We examined efficiency with three efficiency measures: User Turns, System Turns and Elapsed Time. An ANOVA for each of the measures as a function of strategy and task showed that strategy was a significant predictor of efficiency in each case, and that MI was more efficient than SI: User Turns ($F(1,34)=31.9$, $p<.0001$), System Turns ($F(1,34)=14.3$, $p=.0006$) and Elapsed Time ($F(1,34)=3.92$, $p=.05$). Means for these measures are given in Table 3.

Table 3: Efficiency measures for SI versus MI

	SYSTEM (SI)	MIXED (MI)
User Turns	25.94	17.59
System Turns	28.18	21.74
Elapsed Time	328.59 s	289.43 s

Hypothesis H2 concerns the relation between MRS and efficiency. MRS was significantly lower for the MI strategy ($F(1,34)= 27.2$, $p<.0001$), with a mean of .72 for MI as compared with .88 for the SI strategy. The correlation between MRS and Elapsed Time is -.25.

Hypothesis H3 concerned the effect of learning on MRS for the MI interface, and on efficiency as a result. As we hypothesized, MRS did improve as users learned the system ($F(1,70)=6.37$, $p<.01$). The MRS of the MI strategy was .68 for task 1, .74 for task 2 and .76 for task 3, while MRS for the SI strategy was .88 for task 1, .87 for task 2 and .92 for task 3. Furthermore, efficiency was also directly affected by users' learning of the system. There was an interaction between strategy and task for both Elapsed Time ($F(1,70)=4.85$, $p=.03$) and System Turns ($F(1,70)=5.23$, $p=.03$). For the SI system, the mean Elapsed time for task 1 was 321.7s, for task 2 was 345.9s, and for task 3 was 318.2s. In contrast, the MI system started out taking more time on average (task 1=332.6s), but Elapsed Time was reduced significantly over the subsequent tasks, with task 2 taking 302.8s on average and task 3 taking 232.8s. System Turns showed a similar pattern: for the SI system, System Turns for task 1 averaged 26.7, 29.7 for task 2 and 28.1 for task 3. System Turns were reduced for each task for the MI system: task 1 took an average of 25.4 turns, task 2 averaged 22.3 turns and task 3 averaged 17.5 turns.

Hypothesis H4 posited that users' Cumulative Satisfaction (CSAT) for the SI system would be greater for task 1, but

that as users learned the MI system over the three tasks, that the flexibility of the interface and the gains in efficiency would cause MI to be preferred. CSAT was greater for SI for task 1: mean CSAT for task 1 for SI was 27.2 while mean CSAT for MI for task 1 was 23.8. However, despite the fact that the MI strategy was clearly more efficient than the SI strategy by the third task, there was no interaction between strategy and task. There was a significant difference in CSAT as a function of strategy ($F(1,34)= 23.59$, $p=.02$), with mean CSAT being higher for SI (26.6) as compared with MI (23.7). The increases in CSAT for the MI strategy were not significant: mean CSAT for task 1 was 23.7, for task 2 was 22.7 and for task 3 as 24.4. Thus even with the effects of learning, CSAT for MI by the third task was still lower than CSAT for SI on the first task. (See Table 4). Thus Hypothesis H4 is disconfirmed. It appears that, contrary to H4, users' preferences are not determined by efficiency per se, as has been commonly assumed. One interpretation of our results is that users are more attuned to qualitative aspects of the interaction.

To explore this idea further, we first analyzed the relationship between CSAT and our other measures, drawing on the PARADISE framework [6], and its use of multivariate linear regression. We first normalized all measures to their Z scores to ensure that the magnitude of the coefficients in the regression equation would reflect the magnitude of the contribution of that factor to CSAT. An initial regression over a range of measures suggested that Users' perception of task completion (Comp), Mean Recognition Score (MRS) and Elapsed Time (ET) were the only significant contributors to CSAT. A second regression including only these factors resulted in the following equation:

CSAT = .21*Comp + .47 * MRS - .15 * ET

with Comp ($t=2.58$, $p=.01$), MRS ($t=5.75$, $p=.0001$) and ET($t=-1.8$, $p=.07$) significant predictors, accounting for 38% of the variance in R-Squared ($F(3,104)=21.2$, $p<.0001$). This equation demonstrates that while efficiency and task completion are both factors in predicting CSAT, that they are not as significant as MRS. It is plausible that the qualitative behaviors that are correlated with poor MRS have a greater effect on CSAT.

Table 4: Qualitative measures for SI versus MI

	SYSTEM (SI)	MIXED (MI)
MeanRecog (MRS)	.88	.72
Time Outs	2.24	4.15
Barge Ins	5.2	3.5
ASR Rejects	.98	1.67
CSAT	26.6	23.7

This interpretation is supported by measures that more directly reflect the quality of the interaction. See Table 4. First, as discussed above, there were significant differences in Mean Recognition Score (MRS) as a function of strategy. Furthermore, even though users of the MI system were not more likely to ask for help using the always available *Help* command ($F(1,34)=1.47$, NS), they were much more likely to trigger Timeout Prompts ($F(1,34)=10.87$, $p=.002$). Remember that Timeout Prompts are system turns that suggest to the user what they can say, which are triggered by occasions in which the user says nothing after a system utterance. This may happen because the user does not know what they **can** say, or because the user is confused by what the system just did. The mean number of timeouts was 4.15 per dialogue for users of the MI system as opposed to 2.24 for the SI users. Another qualitative aspect of the interaction is the system's production of diagnostic error messages. In our study, it was much more common for the system to reject the utterances of users of the MI system (ASR Rejects), because of low ASR confidence scores ($F(1,34)=4.38$, $p=.04$), leading the system to produce a diagnostic error message asking the user to repeat himself or telling him what he could say. Finally, there was an interaction in the use of BargeIn between strategy and task ($F(1,70)=14.18$, $p=.0003$). Remember that BargeIn may reflect learning; as users learn what they can say, they can barge in over the system's utterances. SI users increased their use of BargeIn over the three tasks, with the number of BargeIns for task 1 at 3.55, task 2 at 4.61 and task3 at 7.33, suggesting that they were learning the interface and becoming more confident. In contrast, users of the MI system started out using BargeIn more (task 1=4.17) but the use of BargeIn decreases with task 2 at 3.89 and task 3 at 2.6. One explanation for the decrease in BargeIns is that users lost confidence in knowing what to say to the MI system.

Other evidence suggests that it was difficult for users to acquire a model of how the MI system worked. Even though the MI system made more errors, users of the SI system were much more likely to use the *Cancel* command ($F(1,70)=18.41$, $p=.0001$), which undoes the effects of the previous command. One plausible explanation of this difference is that SI users acquired a model of the dialogue flow, making it possible for them to use the cancel command effectively, while MI users did not.

Further insight into the factors that affect CSAT can be found by examining the individual satisfaction measures that CSAT is composed of. Users perceive it to be easier to find a message (Task Ease) in the SI condition ($F(1,34)=9.11$, $p=.005$. This is probably because MI users perceived that ELVIS was much less likely to understand what they said (ASR Performance) ($F(1,34)=6.56$, $p=.02$). SI users perceived that the system *often* or *almost always* understood them, while MI users thought the system only

sometimes understood them. As we hypothesized, MI users were more confused about what they could say (User Expertise) ($F(1,34=4.02)$, $p=.04$). MI users were much more likely to say that they only *rarely* or *sometimes* knew what to say, whereas SI users *often* knew what they could say. MI users were also much less likely to say that ELVIS worked the way they expected him to (Expected Behavior) ($X^2=4.6$, $p<.05$). In only 26 out of 54 dialogues did the MI users say that ELVIS behaved as they expected, in comparison with 37 out of 54 for the SI users. This resulted in many fewer MI users saying they would be willing to use ELVIS regularly to access their mail when they are away from their desk ($X^2=4.97$, $p<.05$): in 30 out of 54 MI dialogues users responded *yes* or *maybe* to this question, while the SI users responded *yes* or *maybe* in 41 dialogues out of 54.

Hypotheses H5 and H6 were that users who made use of the summarization and selection by content options provided in both the SI and the MI interfaces would have greater Cumulative Satisfaction (CSAT) and be more efficient than those who chose to listen to their messages in chronological order. In order to test hypotheses H5 and H6, we analyzed the experimental logs and transcriptions for summary use (Suse), next use (Nuse), and content-selection use (Csuse).

Both SI and MI users utilized one of the summarization options that were provided in the system, averaging 1.4 summaries per dialogue. While some users did not use summarization at all, other users summarized up to 5 times in a single dialogue. However, contrary to H5, Suse did not lead to higher CSAT, nor did it lead to greater dialogue efficiency. In fact, our results demonstrate an opposite pattern. An ANOVA with CSAT as the independent variable and Suse as a dependent variable showed Suse a significant predictor of CSAT ($F(5,67)=3.45$, $p=.008$). However CSAT goes down as Suse goes up as shown in Table 5. Furthermore, Suse is highly correlated with Elapsed Time: the correlation coefficient is .49. Thus the more subjects summarized, the less efficient their dialogues were. Analysis of the dialogue transcripts for users who requested the most summaries shows that summaries were used as an error recovery strategy. Users would summarize when the system misunderstood one of the sender or subject values that they had specified when attempting to select by content. Since the sender and subject values were provided in the summary, users would listen to the summary again to make sure that they had specified the values correctly. Thus an increase in the use of summarization indicates a user who was having recognition problems.

Table 5: Cumulative Satisfaction as a function of Summary Use

	SUSE=0	SUSE=2	SUSE=4	SUSE=5
CSAT	26.0	24.97	22.17	18.5

To test Hypothesis H6, we examined the relationship between Nuse (use of the Next command), Cuse (use of the content selection options), and both Cumulative Satisfaction (CSAT) and Elapsed Time (ET). Nuse is not a significant predictor of CSAT ($F(1,106)=3.45$,NS). There was also no effect for Cuse, the use of selection by content options. Further investigation reveals that the main reason for this is likely to arise from the poor performance of ASR for selection by content in the MI condition. In the MI condition, the probability of being correctly understood when using the Read option was only .63. Half of the time if the user specified a selection criteria for reading, e.g. *Read my messages from Owen*, the system misunderstood: sender values were misunderstood 43% of the time. In contrast, MI users who selected messages in chronological or reverse chronological order were correctly understood 81% of the time. Thus there was a great incentive for MI users to **not use** the selection by content options.

On the other hand, if a user chose to access messages by content rather than by order in the SI condition, it took at least three interchanges to say so (see D1). However the overall probability of correct understanding when specifying selection by sender was .76, and for selection by subject was .78. The probability of success for selection by position was .82. Thus there was little difference in the SI condition between system performance for selection by content versus by order.

Table 6: Relationship of Cumulative Satisfaction to use of the selection by Content options in the SI SLI.

	CUSE=0	CUSE=2	CUSE=4	CUSE>5
CSAT	21.0	28.2	26.0	20.0

In order to see whether selection by content is useful when ASR performs appropriately, we analyzed Cuse for the SI strategy alone. An ANOVA of CSAT as a function of Cuse shows Cuse to be highly predictive of CSAT ($F(6,47)=3.89$, $p=.003$). Table 6 shows that CSAT is greatest when Cuse matches optimal performance on the task, i.e. since each task required access to only 2 messages, when Cuse is 2. Thus H6 is disconfirmed for the MI condition, but confirmed for the SI condition.

CONCLUSIONS

This paper evaluates a mixed-initiative (MI) dialogue design in comparison with a system-initiative (SI) dialogue design in ELVIS, a spoken language interface for accessing email by phone. It has been commonly assumed that spoken language interfaces that constrain the user will be less preferred than unconstrained interfaces. Our hypotheses were that users would initially prefer the SI system, which controls the interaction so that the options available to the user are obvious at each point of the

dialogue, and that ASR (automatic speech recognition) would perform better with the SI grammars. However, we hypothesized that as users performed successive tasks, they would learn how to use the MI system, which does not constrain the user, and which is more efficient. We hypothesized that as users learned how to use the MI system, their confidence with the system would increase, and that ASR performance would also increase. Thus, by the end of three tasks, we hypothesized that the satisfaction of MI users would be greater than that of the SI users.

Our results show that the additional flexibility of the MI interface leads to user confusion about their available options and poor performance by ASR. While user expertise and ASR performance did increase for MI over three tasks, these increases did not result in a preference for the MI interface. Despite the fact that the MI interface is more efficient in terms of both turns and elapsed time, the SI users report higher user satisfaction. A multivariate linear regression with user satisfaction as the dependent variable shows that ASR performance, user perception of task completion, and elapsed time are significant contributors to user satisfaction, but that ASR performance is the greatest contributor. We interpret these results to mean that the qualitative behaviors associated with poor ASR performance, the predictability of the system, and the ability of users to acquire a model of system performance, are more important than the commonly assumed performance factors of efficiency and task success.

A user preference for SI style interfaces are also suggested by the results of other work. Previous work has found that directive prompts, and dialogue strategies that structure the interaction with the user, in a similar way to our system initiative interface, are more successful and preferred by users [2,4]. However, in all of these studies, ASR performed worse in the less constrained interface. Future work should examine the role of learnability and predictability with improved ASR performance. In addition, future work should include a longer term study of daily users in the field to determine whether MI interfaces might be preferred by some (expert) users.

REFERENCES

1. Brennan S., and Hulteen E., Interaction and Feedback in a Spoken Language System. *Knowledge-Based Systems*, **8**, (2,3). 1995.

2. Kamm C., User Interfaces for Voice Applications, *Voice Communication Between Humans and Machines.* D. Roe and J. Wilpon Eds. National Academy Press, 1994.

3. Marx, M., *Toward Effective Conversational Messaging*,, MIT Media Lab Masters Thesis, 1995.

4. Oviatt S., Cohen P. and Wang M., Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. *Speech Communication*, 15, (3,4). 1994

5. Walker M., Hindle D., Fromer J., Di Fabbrizio G, and Mestel C. Evaluating competing agent strategies for a voice email agent. In *Proceedings of the European Conference on Speech Communication and Technology*, EUROSPEECH97. 1997

6. Walker M., Litman D., Kamm C. and Abella A. PARADISE: A Framework for evaluating Spoken Dialogue Agents, in *Proceedings of ACL '97* (Madrid, Spain, July 1997), MIT Press.

7. Walker M. and Whittaker S. Mixed initiative in dialogue: An investigation into discourse segmentation. *In Proc. 28th Annual Meeting of the ACL*, pages 70-79. 1990

8. Whittaker S.J. and Stenton S.P., User Studies and the design of Natural Language Systems, *Proceedings of EACL '89, p. 116-123.* 1989.

9. Yankelovich N., Levow G. and Marx M. Designing Speech Acts: Issues in Speech User Interfaces. In *Proceedings of CHI '95* (Denver CO, May 1995), ACM Press.