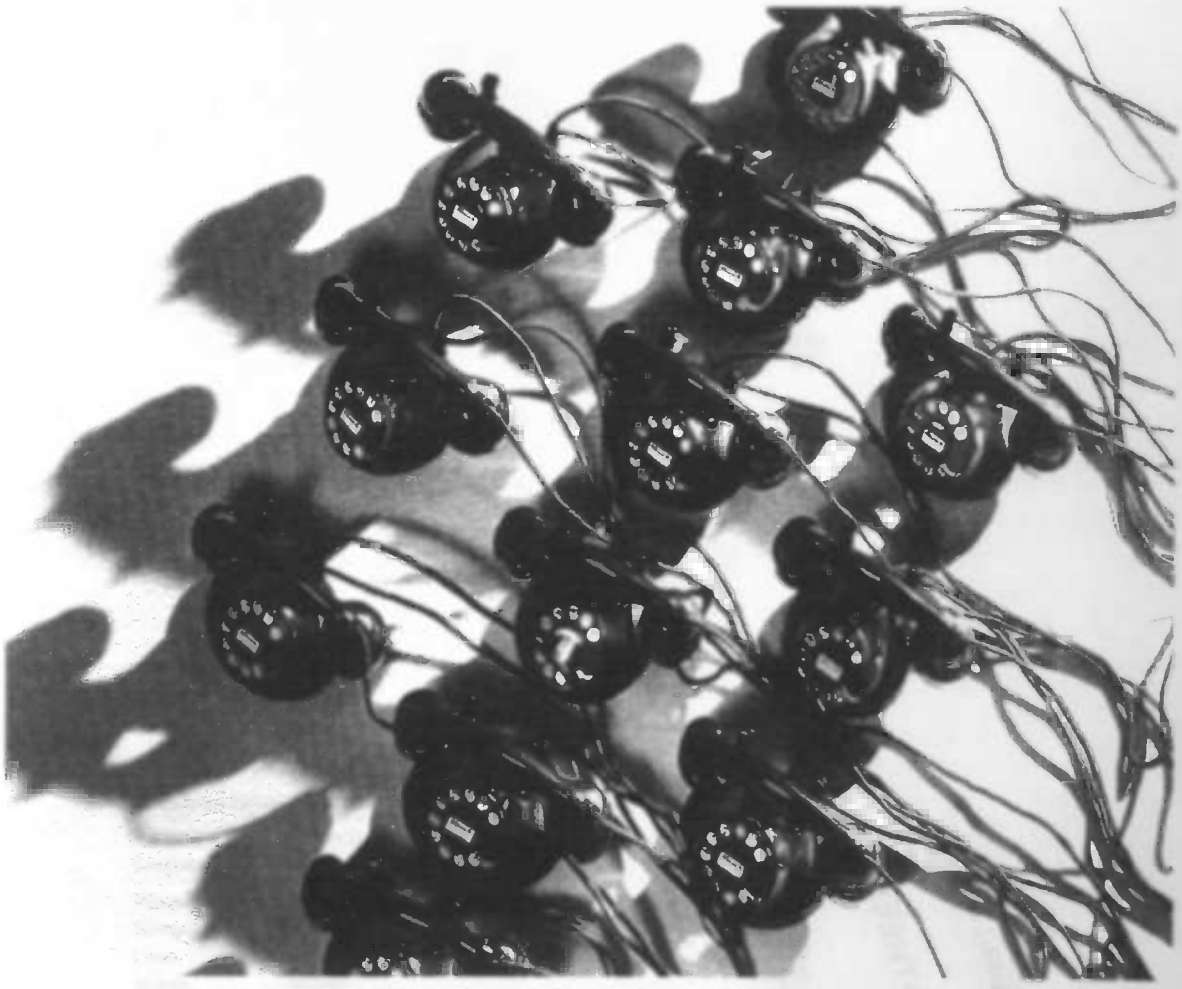
955
1997
005

Prompting for Speech Input in IVR Systems

A Study of User Performance and Acceptance

by

Erik van der Neut



thesis submitted to the faculty of behavioral and social sciences of groningen university in partial fulfillment of the requirements for the master's degree in cognitive science and engineering • this study was conducted at purespeech, inc. and supported by a small business innovation research grant from the national eye institute

m c m x c v i i

968.

Prompting for Speech Input in IVR Systems:
A Study of User Acceptance and Performance

Erik van der Neut

Document Information:

Author:	Erik van der Neut
Language:	U.S. English
Pages:	89
Paragraphs:	1,710
Lines:	4,286
Words:	36,849
Characters:	192,833
Filename:	ThesisReport.doc
Key Words:	ASR, VUI, Human Factors, NEI
File Format:	MS Word 7.0
File Size:	4,410,368 bytes
Paper Size:	Letter 8½" × 11" (21.59 cm × 27.94 cm)

Prompting for Speech Input in IVR Systems
A Study of User Acceptance and Performance

by

Erik van der Neut

A thesis submitted to the Faculty of Behavioral and Social Sciences
in partial fulfillment of the requirements for the Master's Degree in
Cognitive Science and Engineering

This research was supported by a Small Business Innovation Research grant from the
National Eye Institute, Bethesda MD, U.S.A.

Author: Erik van der Neut
Attendant at PureSpeech: G.L. Gabrys, M.Sc.
Attendant at university: L.J.M. Mulder, Ph.D.

Cognitive Science and Engineering
(Groningen University)
Grote Kruisstraat 2/1
9712 TS Groningen
The Netherlands

PureSpeech, Inc.
100 CambridgePark Drive
Cambridge, MA 02140
U.S.A.

Copyright © 1997 by Erik van der Neut

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author of this work must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Erik van der Neut, c/o PureSpeech, Inc., fax +1 (617)441-0001, or erikv@speech.com.

July, 1997

PureSpeech®, ReCite!® and the PureSpeech logo are registered trademarks of PureSpeech, Inc. All other trademarks and service marks are the property of their respective owners.

for Dragana

Abstract

Automatic speech recognition is a promising alternative to touch-tone as a modality for interaction with automated telephone systems. The choices made in the design of speech interfaces will influence the cost-effectiveness and user friendliness of the final product. This report describes an experimental study which was conducted at the speech recognition company PureSpeech, Inc. to evaluate some of these major choices. Different techniques of prompting the caller for input were evaluated with respect to user comfort, task-efficiency, and task compliance. Within-subjects factors were utterance complexity and the presence or absence of input examples, and a between-subject factor was the presence or absence of beep tones as indicators of turn taking. The findings showed that interactions with complex utterances were preferred over simple ones. Systems that did not make use of input examples were preferred over systems that did. Beep tones and input examples increased user performance for complex interactions, but had no effect on simple interactions.

Acronyms

ASR	Automated Speech Recognition
DTMF	Dual Tone Multi Frequency
NEI	National Eye Institute
IVR	Interactive Voice Response
STN	State Transition Network
TTS	Text to Speech
VUI	Voice User Interface

Contents Summary

Abstract iv

Acronyms iv

Contents Summary v

Table of Contents vi

Preface ix

Chapter 1: Introduction 1

Chapter 2: Speech Interfaces for Telephone Systems 3

Chapter 3: Research Question and Rationale 16

Chapter 4: Method 32

Chapter 5: Results 43

Chapter 6: Discussion and Conclusion 50

Appendix A: Implementational Choices 56

Appendix B: Recruitment Poster 62

Appendix C: Prompts Overview 63

Appendix D: Scenario Overview 67

Appendix E: Subjects and Access Codes Overview 68

Appendix F: Subject Instruction 70

Appendix G: Experiment Work Sheet 71

Appendix H: Bipolar Rating Scales 72

Appendix I: Experiment System Preference Form 73

References 74

Index 77

Table of Contents

Abstract iv

Acronyms iv

Contents Summary v

Table of Contents vi

Preface ix

0.1 Cognitive Science and Engineering ix
0.2 PureSpeech x
0.3 National Eye Institute x
0.4 Acknowledgments xi

Chapter 1: Introduction 1

1.1 Objective of this Study 2
1.2 Structure of this Thesis Report 2

Chapter 2: Speech Interfaces for Telephone Systems 3

2.1 Differences between DTMF and Speech 3
2.1.1 Accuracy 3
2.1.2 Level of Abstraction 5
2.1.3 Flexibility 6
2.1.4 Naturalness 6
2.1.5 Ease of Use 7
2.1.6 Costs 7
2.1.7 Conclusion 8
2.2 VUI Design Issues 8
2.2.1 Prompting and Phrasing 9
2.2.1.1 Text To Speech versus Prerecorded Speech 10
2.2.2 Grammar 10
2.2.3 Feedback 10
2.2.4 Error Correction 11
2.2.5 Help 12
2.2.6 Turn-Taking 12
2.2.7 Call-Flow Design 13
2.3 Summary 15

Chapter 3: Research Question and Rationale 16

3.1 System Directiveness 16
3.2 Suitability of ASR for Telephone Information Systems 18
3.2.1 Empirical Support for Usability of Speech 18
3.2.2 Conclusion 20
3.3 User Performance and Acceptance 20
3.3.1 User Comfort 21
3.3.1.1 Sociolinguistic Considerations 22
3.3.2 Efficiency 24
3.3.2.1 Technical Considerations 24

3.3.3 Conclusion 25
3.4 Prompting and User Input Curtailment 26
3.5 Purpose of Experiment 27
3.5.1 Utterance complexity 27
3.5.2 Examples 28
3.5.3 Beep tones 28
3.6 Expected Results 28
3.6.1 Simple versus Complex Utterances 28
3.6.2 No Examples versus Examples 29
3.6.3 No Beep Tones versus Beep Tones 30
3.7 Hypotheses 30
3.8 Summary 31

Chapter 4: Method 32

4.1 Subjects 32
4.2 Procedure 32
4.3 Experimental Design 33
4.3.1 Independent Measures 33
4.3.1.1 Utterance Complexity 33
4.3.1.2 Examples 34
4.3.1.3 Beep Tones 34
4.3.2 Dependent Measures 34
4.3.2.1 User Performance 34
4.3.2.2 User Acceptance 34
4.3.3 Counterbalancing 34
4.4 Materials and Apparatus 35
4.4.1 Four Different Systems 36
4.4.2 Call-Flows 36
4.4.2.1 Error Recovery 39
4.4.2.2 Feedback 39
4.4.3 Prompts 39
4.4.3.1 Prompt Design and Recording 39
4.4.3.2 Feedback in Prompts 40
4.4.3.3 Examples in Prompts 40
4.4.3.4 Overview of Prompt Structure 40
4.4.4 Grammar 40
4.4.4.1 Grammar Size and Recognition Accuracy 41
4.4.5 Eight Different Scenarios 41
4.5 Summary 42

Chapter 5: Results 43

5.1 Performance Data 43
5.1.1 Performance Data Analysis Procedure 43
5.1.2 Analysis of Number of Recognitions 44
5.1.3 Analysis of Total Task Completion Time 45
5.2 Acceptance Data 47
5.2.1 Acceptance Data Analysis Procedure 47
5.2.2 Analysis of Rating Scales 47
5.2.3 Post-Session Interviews 48
5.3 Summary 49

Chapter 6: Discussion and Conclusion 50

6.1 Study Implications 50
6.1.1 Naturalness of Interaction 50

6.1.2 Hybrid Interface Structuring 51
6.1.3 Recognition Accuracy 51
6.2 Study Limitations 52
6.3 Suggested Improvements on the Design of this Study 53
6.4 Suggestions for Future Research 53
6.4.1 Hierarchical Utterance Complexity 53
6.4.2 User Sensitive Prompt Adjustment 54
6.4.3 Sociolinguistics 54
6.4.4 'Barge-Through' 54
6.4.5 User Instruction and Error Correction 54
6.4.6 Flexibility of Speech Interface Design Toolkits 55
6.5 Summary and Conclusion 55

Appendix A: Implementational Choices 56

A.1 Four Different Systems 56
A.2 Call-Flows 56
A.2.1 Error Recovery 57
A.2.2 Feedback 58
A.3 Grammar 59
A.3.1 Grammar Size and Recognition Accuracy 59
A.4 Prompts 59
A.4.1 Design and Recording 59
A.4.2 Feedback 60
A.4.3 Examples 60
A.5 Scenarios 61

Appendix B: Recruitment Poster 62

Appendix C: Prompts Overview 63

C.1 Commuter Rail System 63
C.2 Auto Loan Calculator 64
C.3 Auto Blue Book 65
C.4 Catalog Order System 66

Appendix D: Scenario Overview 67

Appendix E: Subjects and Access Codes Overview 68

Appendix F: Subject Instruction 70

Appendix G: Experiment Work Sheet 71

Appendix H: Bipolar Rating Scales 72

Appendix I: Experiment System Preference Form 73

References 74

Index 77

Preface

For the graduation of my Master's course in Cognitive Science & Engineering, I worked for seven months as an intern at the speech recognition company PureSpeech, Inc., on a project supported by a grant from the National Eye Institute.

Major technological developments, from the personal computer and the micro-wave, to the cellular phone, etc., have led people in the past two decades to depend increasingly on their interaction with machines. The design that shaped this machinery or software has mostly been affected by the limitations and possibilities of the technology. Because most of those designs were not guided principally by human capacities and limitations, the human-machine interaction has not always been satisfactory.

In recent years more attention is being directed at potential users of new technology. Out of this tendency rapidly evolved a need at the companies that provide such technology for skilled engineers with a good knowledge of Cognitive Psychology.

0.1 Cognitive Science and Engineering

In the early Nineties, it became clear to a small group of professors and lecturers at the Groningen University that there was a need for a degree course that combined Cognitive Psychology with Engineering aspects such as Computer Science and Physics. Problems with the existing interdisciplinary contacts between researchers, mainly in the United States, ranged from differences in jargon to differences in fundamental research questions. Educating and training a new generation of researchers capable of overlooking the entire interdisciplinary specialty seemed like a way to overcome this problem. In the prospect of providing such a solution, the degree course of Cognitive Science and Engineering was introduced in 1992 at Groningen University, in the Netherlands.

Cognitive Science and Engineering deals with human knowledge and mental processes. The architecture, functionality, and limitations of the human brain play an important role in practically everything that people do. Understanding mental capabilities and limitations, and understanding human behavior in relation to technology are the main goals of Cognitive Science and Engineering. The Human Factors community believes that this knowledge will allow designers of new technology to make their products easier to use, while providing models for building cooperative or intelligent technology.

Contemporary developments in information technology provide an applicability in the engineering field for theoretical models from Cognitive Psychology, Linguistics, Logic, and Philosophy. In addition to these scientific areas, Cognitive Science and Engineering also includes Computer Science, (Bio-)Physics, and Neuro-Science. The necessary interdisciplinary specialty was created by integrating those different fields in a single curriculum.

I had the challenging pleasure of being a member of the small group of students that made up the first generation to read Cognitive Science and Engineering at

Groningen University. To conclude an exciting study period with an equally exciting project, I worked as an intern for seven months at the speech recognition company PureSpeech, Inc.

0.2 PureSpeech

At around the same time Cognitive Science and Engineering was introduced at Groningen University, another group of people had a vision and a goal related to these new technological developments. Benjamin Chigier founded Integrated Speech Solutions in 1992 in Boston, Massachusetts. Chigier received his training in Speech Technology from Carnegie-Mellon University, and has worked for the Speech Technology Group of NYNEX Telecommunications. His ambition was to transfer advances from the research community to the commercial market, and provide speaker independent continuous speech recognition systems for use in practical applications.

His company—now named PureSpeech, and based in Cambridge, Massachusetts—builds natural voice user interfaces (VUI) and is making a special effort to deploy highly accurate and cost-effective speech recognition solutions for the high call volume Computer Telephony market. Based on the results of existing earlier research in speech recognition, speech processing, statistical modeling, language modeling, natural language processing, and human factors engineering, PureSpeech specifically designs its own products, rather than licensing existing technologies. With this approach, PureSpeech deploys a software-only and Digital Signal Processing (DSP) based solution for the telephony market.

ReCite!, PureSpeech's suite of Automatic Speech Recognition (ASR) products, is a toolkit for building Speech Recognition interfaces. It features speaker independent, continuous speech recognition, and achieves accuracy levels of 96 to 99 percent for large and constantly increasing vocabularies under ideal circumstances. PureSpeech is currently working on speech interface modules for the ReCite! toolkit. These modules are software objects that perform pre-configured interactions.

Human Factors engineering is a very important aspect of the design at PureSpeech. PureSpeech's goal is to build systems with which users can interact in a natural manner. Insights from Human Factors are essential for developing 'the natural interface'.

My work at PureSpeech involved Human Factors research on speech interfaces for telephone information systems. The study is described in this report.

0.3 National Eye Institute

The National Eye Institute (NEI) is funding a Small Business Innovation Research grant to PureSpeech for a project to provide telephone-based wayfinding in a transit information system to be used by visually impaired as well as sighted people. The study described in this report is part of that project.

The NEI is a department of the National Institutes of Health in Bethesda, Maryland. The NEI conducts and supports research, training, health information dissemination, and other programs, with respect to visual impairment and the special health problems and requirements of the blind. Over 85% of its appropriated funds is used to support extramural research and research training at universities, medical schools, hospitals, and other institutions in the United States and abroad.

Blasch & Hiatt (1983; as referred to by Chigier, 1996) suggested speech as an input medium for wayfinding information for persons who are visually impaired. The experiment described in this report was part of the second phase of this project that was concerned with enhancing a prototype of such a system so it would meet the needs of its

users in the real world. Enhancing the prototype would be achieved by enhancing the robustness of the recognition system, collecting additional data to train the speech and natural language components of the system, and by conducting usability studies to enhance the user interface.

0.4 Acknowledgments

Without the great deal of help and guidance that I have received from a large number of people, this graduation report would not have been what it is now. My colleagues at PureSpeech, people at the university, as well as friends and family, have been very supportive throughout my internship.

At PureSpeech I would like to thank Gareth Gabrys for his supervision and his efforts in making this first experiment of the National Eye Institute project a valuable Master's thesis study, Shelly Dews for believing in me and getting me to the company, Paula Kirtley for reading on disk the 302 prompts for the experiment application, Diane Ballestas for helping test the experiment application, Mark Pundsack and Amy Limb for providing me with technical information, and all my colleagues for the very special and wonderful working atmosphere.

At Groningen University I would like to thank Ben Mulder for being my attendant, Annemieke Galema of the Internship Bureau of the Faculty of Arts for her exceptional efforts to obtain a United States visa for me, Tjeerd Andringa for introducing me to the world of commercial speech processing applications, my professors, lecturers and fellow students for making Cognitive Science and Engineering such an interesting and dynamic study, and many other people for their advice during my preparations.

Of my friends and family I would like to thank everybody for their support and interest. A very special thank you goes to Dragana Miljkovic, my wife-to-be, who has spent a great deal of her time on reading draft versions of this report and giving useful suggestions for improving the writing. With that and her tremendous support she has made an invaluable contribution to this work.

1 Introduction

A short explanation of IVR systems, the objective of this study, and the structure of this thesis report

With the tremendous overall technological developments in recent decades, the use of Interactive Voice Response (IVR) systems has increased dramatically. IVR systems are automated telephone systems providing information or a service. They are accessible over the public telephone network and are completely auditory. Applications varying from airline reservations to banking, and from train schedule information to filling out tax returns, use these fully automated telephone systems to provide information or receive input from their callers. The need for efficiency has eliminated the human operator from these interactions, usually providing a touch-tone interface instead. Complex and time-consuming human discourse is replaced by an abstract, fully controllable menu-structure.

Touch-tone, also known as Dual Tone Multi Frequency (DTMF), interfaces have made many services available over the ordinary telephone line. A DTMF interface is able to recognize the digits of a touch-tone telephone. By playing recorded speech that prompts the caller to press certain keys for certain options, and by making choices based on the digits it recognizes, a touch-tone interface can guide the caller through a menu structure. Different prompts will be played according to the route chosen through the interface structure. Information can be provided by playing recorded speech, while services can be provided by triggering actions in other applications according to the touch-tone input.

Contemporary automated telephone systems primarily use DTMF menu structures for their interface. The advantages of these IVR systems are that virtually anyone can use a touch-tone telephone, and has access to one. By running the application on a system that can handle multiple phone lines, and by taking callers through a short and straightforward menu, these systems can handle a vast number of users in a short period of time. IVR systems also have many benefits for businesses, such as reducing personnel costs, and offering more services at practically unlimited hours.

Recent advances in Automated Speech Recognition (ASR) open the doors for an even greater variety of IVR systems. An ASR interface accepts natural speech as its input. With a set of algorithms and rules for, among others, syntax and semantics, it parses the input wave form and builds a model of the spoken phrase. It will step through the interface structure, play auditory prompts to the user, and perform other actions based on the meaning of the words it recognizes. PureSpeech[1] focuses entirely on providing speech solutions for the telephony market in its belief that speech is the most natural way of communication, and that it can be used to make human-computer interfaces more natural as well.

[1] Please see the preface for an elaboration on PureSpeech, and its relation to this project.

1.1 Objective of this Study

The market for IVR systems is still expanding. The increasing need for automated telephone information systems has made the IVR market a serious business, and owners of such systems want their investment to be cost-effective.

With current developments speech recognition is increasingly used as a modality for the interfaces of IVR systems. The implementation of ASR for automated telephone systems is not a trivial task. Cost-effectiveness of the system is influenced by the users' acceptance of the system and their performance. When ASR is used for the interface, it needs to be both user friendly and efficient.

The purpose of this study was to examine the influence of certain aspects of the design of speech interfaces on user performance and acceptance. The study focused particularly on different techniques for prompting.

1.2 Structure of this Thesis Report

To clarify the field of the research described in this report, the *Introduction* elaborated shortly on IVR systems, DTMF, and ASR.

Chapter 2, *Speech Interfaces for Telephone Systems*, describes speech as a modality for automated telephone systems. How speech can overcome limitations and problems encountered with touch-tone interfaces is outlined in a comparison of ASR and DTMF. How speech can be implemented in interfaces is described with reference to speech interface design issues.

The structure of the rest of this report is similar to the composition of a research article, with the research question and rationale, description of the method, study results and discussion, and the conclusion in separate chapters. The appendices include all documents used for the design and execution of the experiment, such as call-flows, subject recruitment posters, scenarios, etc.

Chapter 3, *Research Question and Rationale*, provides empirical support for the usability of ASR in IVR systems, elaborates on user performance and acceptance with reference to sociolinguistic and technical considerations, and describes the influence of different prompting techniques on the interaction between an ASR system and its users. The purpose of this study and the expected results are explained here as well.

Chapter 4, *Method*, describes the subjects of this study, the procedure, the experimental design, and the materials and apparatus used. All choices relevant to the experimental design are covered in this chapter. Important choices not directly related to the design, such as consistency considerations, are described in *Appendix A: Implementational Choices*.

Chapter 5, *Results*, describes the data analysis and study results. User performance data and user acceptance data are described and discussed, and observations from the post-session interviews are outlined as well.

Chapter 6, *Conclusion*—the last chapter, explains the limitations of the study, evaluates the study implications, suggests improvements on the experimental and overall design of this study, and proposes further related research.

2 Speech Interfaces for Telephone Systems

A comparison between touch-tone and speech as a modality for interaction, and an elaboration on speech interface design issues

DTMF is increasingly applied in IVR systems, and users have no choice but to accept the wide variability in quality of these interfaces. Schumacher, Hardzinski & Schwartz (1995) speak of widespread dissatisfaction caused by poorly designed IVR systems. In the ASR community it is widely acknowledged that a way to change such dissatisfaction would be to improve user comfort and efficiency, and that this may be achieved by using speech as the modality for the telephony interface instead.

Such a change may not come easily, as automated telephone systems, dominated by touch-tone interfaces, have already proven their usability and usefulness. Having made it possible to develop and deploy fully controllable automated systems that can handle a vast number of callers at every hour of the day with an interface that virtually everyone can use, DTMF has set a clear benchmark. In order for a speech interface—also known as Voice User Interface (VUI)—to prove itself a better alternative, it has to measure up against the advantages of DTMF.

The ASR community believes that speech interfaces can overcome the limitations imposed by touch-tone interfaces. If simply telling the IVR system what is desired indeed makes the interaction more intuitive, more straightforward and faster, then speech will be a very suitable and important modality for IVR systems.

The comparison between DTMF and ASR in the next section, *Differences between DTMF and Speech*, will provide further insight. A set of design issues are of importance with respect to the implementation of a speech interface. These issues will be outlined in section 2.2, *VUI Design Issues*.

2.1 Differences between DTMF and Speech

There are a number of important differences between telephone interfaces that rely on DTMF and those that are based on speech. In the following paragraphs, the comparison of these differences with respect to accuracy, abstraction level, flexibility, naturalness, ease of use, and costs indicates the disadvantages and advantages of speech over DTMF. If the disadvantages can be minimized and the advantages can be optimized, then speech could become the better candidate for many telephone interfaces.

2.1.1 Accuracy

One essential difference between DTMF and speech is that DTMF is more accurate. Touch-tone input is almost always perceived correctly by the system, while speech recognition still has a considerable failure rate. Factors that make ASR complicated and less reliable than DTMF are variations in the way the speech sounds are pronounced, variations in background noise, and variations in what is being said.

The variation in speech sounds should be handled by the robustness of the recognizer—perhaps in combination with socio- and psycholinguistic knowledge—and noise should ideally either be filtered out or reduced. The variation of what is being said to the system cannot be dealt with by the recognizer only, however. Natural language

Figure 2.1—"...speech recognition still has a considerable failure rate."

leaves room for many different ways of formulating the input and a speech recognizer may not be able to handle such variety. For a speech interface to be as effective as DTMF in classifying user input, it is the combination of a robust recognition engine with an interface compensating for the limitations of the recognizer that can overcome this problem of versatility of natural speech.

Zoltan-Ford (1989) describes two solutions to this problem. The first one is to program the computer to recognize and understand the many ways people can structure their inputs. This solution is rather obvious, but is not a likely option for domains with a high degree of complexity. The problem with this option is that, when users are allowed a lot of freedom, the range of possible words and phrase structures used for the input increases dramatically. The range of possible input that a recognizer is able to capture is specified in a so-called *recognition context* for every state of the interaction. Large recognition contexts hinder accurate speech recognition. Even with a strong speech recognition engine, trying to capture all possible input would be a very inefficient approach. While a robust recognizer will always be important in this respect, it is the design of the interface that will have a direct effect on the utilization of the strength of the recognizer. The interface can contribute to the realization of speech as an accurate modality for input that makes for an efficient and pleasant interaction between a system and its users.

The second solution described by Zoltan-Ford (1989) indicates how the interface can make such a contribution. It involves curtailing the variability of what users say to the system. Inducing users to adjust their input to the limited understanding of the system can be done either openly or in a more discreet manner. The overt approach would be, for instance, to present the users with a list of acceptable commands during the session. A covert approach would be to let a limited set of words and phrases used by the system itself serve as a model for the users' input. The system output uses words and phrase structures that it expects for the input it has to recognize. This way, the output of the system serves as an implicit template for users to incorporate in their choice of words and syntax for their communication with the system. The covert approach might be accomplished by taking into account empirical findings about human conversations, and translating them into Human Factors guidelines for setting up the interface. Conversational cues about turn-taking and feedback, and so forth, are unobtrusive in inter-human conversations, and lead to very efficient discourses (Engel & Haakma, 1993). Whether the system's efforts to reduce the variability in user input should be explicit or more implicit to the user is a choice that depends on the type of application.

By working on these two solutions simultaneously—adjusting the computer to the user, *and* vice versa—as much as possible, the advantages of speech can be optimally utilized within an interface. In this way, contemporary limitations of the speech recognition technology can be overcome by proper design of the speech interface.

2.1.2 Level of Abstraction

A second difference between DTMF and speech lies in their abstraction level. While touch-tone interfaces have digits assigned to different options—something that often seems to be done arbitrarily—a speech interface allows the user to name any option directly. Unlike speech commands, digits in themselves do not convey the meaning of the choices they stand for. This difference has two important consequences. The first is that it can be very cumbersome to navigate through a DTMF menu structure, and the second consequence is that with DTMF it is a tedious procedure to make a choice from a long list of options.

When interacting with a typical touch-tone based IVR system, users that do not know the options and matching digits by heart have to wait and listen to a list of options at every step until the desired option is mentioned. If the users are not sure of the way the options are organized in the touch-tone tree, then they may even want to hear all options at every step to make sure they have chosen the most appropriate option. At the same time, the callers have to remember the matching key-pad numbers for options of interest, before actually deciding which one to choose. The following transcription of a call to the American Consulate General in Amsterdam illustrates such a structure:

> ACG: "If you would like information about visa to visit, work or live in the United States, press '1'. For information about services for American citizens such as Passport or Social Security information, press '2'. For information about our trade services, press '3'. To speak to the operator, press '4'. To return to the main menu of choices, press '5'."
> *Caller presses '1'.*
> ACG: "If you would like information about traveling to the United States for a vacation or a short business trip, press '1'. For information about visa to do a study or do an internship in the United States, press '2'. For information about visa to work temporarily in the United States, including as an au-pair, press '3'. For information about immigrating to the United States, press '4'. To return to the main menu of choices, press '5'."
> *Caller presses '3'.*
> ACG: "In some cases foreigners are allowed to work in the United States. For general information about working in the U.S., press '1'. For information about going to the U.S. as an au-pair or summer camp councilor press '2'. For information about visas for investors or traders, press '3'. To return to the main menu of choices, press '4'."
> *Caller presses '1'.*
> ACG: *Detailed information is provided about the conditions for and the procedures of getting a non-immigrant worker visa, directly followed by:* "If you would like to have information about temporary worker visas back to you, press '1'. To repeat this message, press '2'. To return to the main menu of choices, press '3'. To speak to an operator, press '4'."

Figure 2.2 shows the DTMF tree of the above transcription. The route through the menu structure followed in the transcription is indicated by boxed menu options and bolded user input.

Depending on the number of options and the structure of the touch-tone tree, such an interface can be mentally highly demanding. In a speech interface it is less likely that the lists of options need to be heard completely or at all, since the lack of abstract matching makes the commands much more intuitive. Also, when the user is unfamiliar with the system and wants to hear all options, it is easy to remember the desired command. In the above example, for instance, if the user needs information about work visas then with a speech interface they could simply say "Work visa" after an opening

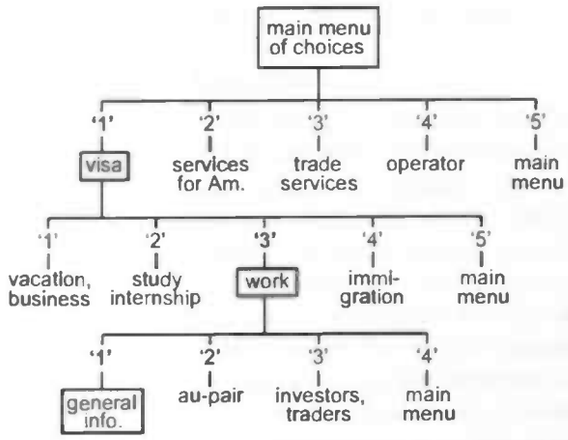


Figure 2.2—Touch-tone tree representation of part of the menu-structure of the IVR system of the American Consulate General in Amsterdam

prompt such as: "Would you like information about visas, about services for American citizens, or about trade services?" If they would just say "About visas", then the system could ask for the type of visa.

Some IVR systems let the callers specify a name from a list of options, for instance to determine the destination of a flight. As we have seen, touch-tone systems are inadequate when it comes to making a choice from a long list of options. In addition, spelling out names on the touch-tone key-pad is a tedious process that can be downright frustrating. This makes speech the better modality for systems that require their users to specify names—of persons for example, or a stock name, or the names of the railway stations of a journey. Such systems implemented in DTMF are not only user unfriendly, but are also bound to be time-inefficient.

2.1.3 Flexibility

A third difference between DTMF and speech is that speech can offer much more flexibility in an interface than DTMF. Flexibility in this respect means that the user has the option of specifying an action, or entering or requesting information in a *single* query, where a DTMF interface would need a deep hierarchical structure to lead the user to the desired action. More flexibility can be translated into increased efficiency and greater user comfort since it allows users to conduct the interaction at their own pace. Speech is more efficient since it leaves room for a limited set of high level commands that can be used in a shorter time frame than a longer series of steps that would be necessary with DTMF. Greater user comfort results from these commands being straightforward and intuitive. This is especially the case when the parameters of those commands can be specified directly as well, since it leads to an interaction with a level of efficiency that cannot be achieved by DTMF interfaces. A single speech command can span several levels of a touch-tone tree. For instance, instead of pressing a '2' to make a call, then pressing '3' to pick Amy, and then pressing '2' again to reach her at work, a speech interface would allow the user to simply say "Call Amy at work."

In addition to higher user-input flexibility, speech has the advantage of also being able to serve callers using a telephone that is not DTMF compliant. A higher degree of user-hardware flexibility is invaluable outside the Western world especially, where not all phones are DTMF compliant.

2.1.4 Naturalness

A fourth difference between DTMF and speech is that speech is a natural way of communication among people, while a telephone key-pad is not. Although touch-tone has become quite natural for IVR systems, since people have become familiar with it, a speech interface makes use of communicative skills that people have developed throughout their lives. When the interface is designed in such a way that the users intuitively know what to say, then anyone who speaks a natural language that is the same as the language spoken by the system, can interact with that system. PureSpeech

especially has made it a point to work on the user interface having a natural appearance and functionality to the user.

2.1.5 Ease of Use

A fifth difference between DTMF and speech, resulting directly from the second, third and fourth difference described above, would be ease of use. While in a DTMF interface commands are mapped to arbitrary choices on the abstract telephone key-pad, a speech interface allows the user to enter commands or data by using the actual and natural names. This makes speech commands easier to think of. It also makes the commands easy to remember. Commands of a DTMF system that has been used very frequently will probably be remembered easy as well. When a touch-tone interface allows for making choices before they are mentioned as an option, users can quickly get where they want by pressing a memorized sequence of digits.

Studies have been conducted on ease of use of speech interfaces. Zoltan-Ford (1989) examined to what extent people can be shaped to conform their input to the syntax and vocabulary used in the output of an inventory program. In this study, in which users were told that the program was capable of recognizing natural-language input and keyboard input, there were more messages by voice input than by keyboard input, showing that users preferred speech over typing. A larger number of voice input messages, however, may also indicate an uncertainty on the part of the users about the computer's understanding of their voice input. The data supported the claim, however, that voice input was simply easier than keyboard input.

Users enter requested data or queries when interacting with an IVR system. For these kinds of tasks, natural speech is an effective method of interaction (Capindale & Crawford, 1990). Extra memory load will result, however, from the required feedback that needs to be transferred over the same auditory modality. A certain amount of redundancy is necessary with natural language because of its low information density. Intuitively, listening and speaking at the same time may therefore be difficult because of limited human auditory resources, which would make a solely auditory speech interface hard to use. However, in a study of dual-task performance Shallice, McLeod & Lewis (1985; as described by Wickens, 1992) found that the human resources underlying speech perception and speech production are separate. According to the Multiple-Resource Theory (Wickens, 1992) this means that listening and speaking go well together. This is true for tasks, such as simultaneous translation, in which the information content of both listening and speaking is similar. When listening and speaking are dealing with more different information, however, as in question-answering tasks in IVR systems, then the working memory is the limiting factor. This means that a sequential alternation of listening and speaking has to be implemented in speech interfaces of IVR systems.

That speech commands seem to be easier to use than keying, and that the necessity of feedback over the same modality does not seem to pose problems with respect to human resources, indicates that speech can make an IVR interface easier to use. This is an important issue, since systems that are easier to use will either be more efficient, or will have a wider range of users, or both.

2.1.6 Costs

A sixth difference between DTMF and speech is the cost of implementation and deployment. Speech interfaces are more expensive than DTMF interfaces. Recognition of the DTMF tones is trivial and commonly available on inexpensive line cards and modems. Additionally, the DTMF interface design is fairly straightforward, it can be made quickly even by beginners and many development kits are available for it on the market. Speech

interfaces are far more complicated to build and development kits are only beginning to become available. In addition to the specially developed software, speech recognition commonly requires expensive DSP hardware.

This is an important disadvantage of speech interfaces, since telephony companies will be less willing to deploy a system with high startup costs. To make speech interfaces financially viable, the initial startup costs must be offset by savings over the longer term, due to more efficient interaction. Therefore, a speech interface has to deliver much more efficiency and throughput than a DTMF interface in order to be a suitable substitute.

2.1.7 Conclusion

In conclusion, speech offers many advantages over DTMF as a modality for a user interface. It is much more flexible, more natural, and makes properly designed interfaces easier to use. On the other hand, DTMF is more direct, and less expensive. Further work on Voice User Interface design should reveal ways to minimize the uncertainty of recognition results, and make speech a cost-effective substitute in telephone information systems.

2.2 VUI Design Issues

The acceptance of ASR in Computer Telephony will depend as much on good Human Factors design as on the accuracy and efficiency of the recognition engine. The importance of Human Factors research on speech interface design was confirmed in two problems that emerged from a study conducted by Brems, Rabin & Waggett (1995). In their four experiments—which examined the usability of natural language conventions for improving speech interfaces—subjects called in to an automated telephone operator service that worked with ASR, and simply responded to questions from the system. The researchers found that user commands were often embedded in more complex utterances, and that prompts needed to be long in order to be able to instruct users how to phrase their input. The problem with complex utterances is that they are difficult to recognize, and the problem with long prompts is that they become irritating to the user (Brooks, 1989; as referred to by Brems *et al.*, 1995). Also, longer prompts require prolonged attention.

Robust recognition is essential since misrecognition makes an application inefficient. Extra time is needed to correct the errors, and the necessity for an error recovery mechanism itself may introduce time-inefficiency, by the need to explain its functionality or name it as an option. In addition, users have a very low tolerance for low recognition accuracy (Casali, Williges & Dryden, 1990). This means that the two problems found by Brems *et al.* (1995) need to be overcome in order to make ASR more widely applicable in real-life systems.

The problem of user input being too complex to recognize can only partly be solved by the use of more computing power and special mechanisms such as word spotting[2]. Continuously improving the speech technology is not likely to be sufficient to handle the variability of natural language in the near future. Zoltan-Ford (1989) proposes that users be prevented as much as possible from embedding their input in extraneous speech. This can be achieved by using cognitive psychological and linguistic knowledge for the design of the speech interface. Knowledge about users' intentions, their mental

[2] The mechanisms of word and phrase spotting allow for the recognition of key-words and key-phrases in extraneous speech. A recognizer equipped with such mechanisms scans the speech input for the words or phrases that the system can recognize. The parts of the input that cannot be matched are disregarded. With this approach it can detect "From Chicago to Boston", for example, in a phrase such as "Uhm... I'd like to fly from Chicago to... uh... Boston."

model of the interaction, and their need for feedback can contribute to the design of an interface structure that requests the user for information at intuitively correct moments. Linguistic knowledge on how conversational partners keep track of exchanged data and the status of a dialogue, can propose guidelines for the formulation of the system output. It can also facilitate the design of the specification of what the system can recognize. With a speech interface that asks the right questions and has a solid expectation of the input, users are less likely to say out-of-set phrases.

Human Factors techniques can also be used to find a solution to the necessity of lengthy prompts explaining how to phrase the input. A way to do this might be to design implicit prompting in such a way that it has a high chance of eliciting the right response in a certain format. Conversational partners frequently and unknowingly adopt each other's conversational style in human dialogue (Danzinger, 1976; as referred to by Zoltan-Ford, 1989). Assuming that a certain format of the system output increases the likelihood of users formulating their input similarly, users may automatically adjust their input in the desired way when system output resembles the desired input with respect to syntax and semantics.

It will need good interface crafting to take advantage of the faster speed of speech. Only a carefully designed speech interface can overcome many aspects of the problematic combination of the complexity of speech and the current technological limitations. Even when technology is not fully capable of dealing with the complexity of its domain, it should be able to produce an outcome that is successful from the user's point of view. A number of issues need to be taken into account in the design process of such a speech interface. A closer look at them will provide an overview of the domain of speech interface design.

2.2.1 Prompting and Phrasing

High levels of speech recognition accuracy are still very difficult to attain. Lea (1982; as referred to by Casali *et al.*, 1990) suspects more than eighty variables influencing speech recognition accuracy. Since we are dealing with fallible speech recognizers[3], the main question is how to engender users to produce the input that the system can recognize. The most direct and important influence on how callers phrase their input, is the way they are prompted for it by the system.

Prompts are output by the system that either give the user instructions or information, or ask the user for a particular input. There are many different ways of prompting, of which the prompts that ask for input are of primary importance to phrasing by the user, and therefore to efficiency. In an internal study at NYNEX Telecommunications significant differences were found in subjects' task compliance as a result of a prompt-type treatment (G.L. Gabrys, M.Sc., personal communication, November 1996, PureSpeech, Inc., Cambridge, MA). The phrasing of the input, as expected by the system, will set a minimum requirement for the grammar that specifies what the system can recognize.

Different ways of prompting can result from varying and combining several prompting techniques such as the length of the prompt, the use of examples, and the use of beep tones as speaking cues. Prompt length is assumed to have the effect of eliciting user input of the same length and comprehensiveness. Examples can clarify how the user can phrase the input, but they also slow down the interaction. Beep tones can be used as

[3] PureSpeech has recently built a recognizer that is able to achieve high accuracy levels, in the order of 96% to 99%. Even if recognizers were perfect however, they would not be able to recognize out-of-set data.

an indication of turn-taking, to prevent users from talking while the system is not yet ready to process their input.

2.2.1.1 Text To Speech versus Prerecorded Speech

Speech output can be generated by a Text To Speech (TTS) engine. TTS is synthetic speech. A TTS engine takes a text string as input and produces speech sounds according to a set of transformation rules. Another solution is to record a human voice, and string the prompts together from different wave files. Using synthetic speech is much easier than making all the separate recordings of prompts or parts of prompts and programming them to be stringed together. Synthetic speech does not sound natural, however.

There are also empirical findings that TTS has a negative influence on the performance of the users. Ralston, Pisoni, Lively, Green & Mullennix (1991; as described by Paris, Gilson, Thomas & Silver, 1995) concluded that processing speed and comprehension of natural speech was better than that of synthetic speech. It must be noted, however, that the TTS engine used in their study was relatively poor. Synthetic speech systems do not have the same richness as natural speech (Spiegel, Altom, Macchi & Wallace, 1988; as referred to by Kamm, 1994), and the perception of artificial speech imposes greater processing demands on the listener (Luce, Fuestel & Pistoni, 1983; as referred to by Kamm, 1994).

Applications that must be able to generate an unlimited vocabulary, such as reading machines for the blind, for instance, usually make use of TTS (Paris *et al.*, 1995). When the vocabulary of an application is limited and all possible speech output is fully known and stable over time, prerecorded speech is an appropriate choice for the voice output (Kamm, 1994).

2.2.2 Grammar

The interaction between a user and an ASR interface consists of a series of steps or states. At each state there is a certain set of words or phrases that the system can recognize. This set is called the *recognition context* and is defined by the grammar. The grammar specifies the words—commands, for example, the type of words—city names, for instance, and phrases—such as queries, that the system expects in the user input at every step. Anything that is not contained in the grammar cannot be recognized.

Since the system's requests for input are supposed to elicit a certain range of responses, the grammar needs to cover this range. At every state both the grammar and the prompts must refer to the same domain, and imply the same syntax to the extent necessary. For example, a very open-ended prompt may not be very helpful when the grammar is restricted. When an opening prompt of a transit information system asks users to specify their journey, the grammar needs to include all words and phrases that can be expected in the response to this request, such as station names, times, and dates, for instance. Since users may embed their input in extraneous speech, which mechanisms such as word spotting can only partly allow, it is better to let grammars cover a somewhat wider range than the necessary input. By letting the grammar be a superset of the domain implied by the prompting, it will have a high chance of containing the actual user responses.

2.2.3 Feedback

Feedback from the system to the user is of special importance in a speech interface. Capindale *et al.* (1990) observed nineteen volunteers interacting with a Student Information System which used a natural-language interface. For the users of this database application, feedback reduced both training and start-up time. Feedback also

Figure 2.3—"Feedback from the system to the user is of special importance in a speech interface."

maintained the flexibility of the natural language interface, and it worked towards motivating the users. In addition to error messages, the feedback consisted of an echo of the user input. This echo was a translation from the natural language to a database query and appeared on the screen in front of them. The participants felt that the feedback taught them how to phrase subsequent requests and it encouraged them to explore a wider variety of queries.

In an elaboration on the 'layered-protocol model' of Taylor (1988), Engel *et al.* (1993) speak of the necessity of display of feedback and expectations in user interfaces. They draw the conclusion that, in a user interface, feedback, both about interpretations and expectations, as well as correction procedures, needs to be presented to the user in order to make the interface efficient, and easy to use. Feedback allows for the communication of intentions, and for verifying whether messages are correctly interpreted. Efficiency of communication is a determinant of user comfort (Casali *et al.*, 1990) and user performance.

Feedback is essential, but at the same time it is problematic in a speech interface because of the alternation of listening and speaking by the users. Since for interactions with IVR systems both user input and system output are transferred over the same channel, namely the auditory modality, feedback cannot be presented at the same time as other output or during user input. Even if feedback could be played during user input, it would be highly demanding for the user to formulate input while listening to the output. This means that feedback has to be carefully placed in the sequence of input and output phrases.

2.2.4 Error Correction

It is important for every type of interface to provide for error correction. This is especially true when costs induced by errors are high, for example when funds are transferred between bank accounts. Errors can result from users saying the wrong thing, or the system misrecognizing what they say. Speech recognition will succeed with a certain probability, which is most often around 90% for in-set utterances for

contemporary commercial applications[4]. For the remaining cases, along with out-of-set data and user mis-statements, an error correction mechanism is needed.

There are several possibilities for such a mechanism. One option that is often used is explicit confirmation. The system tells the user what it recognized, and asks for confirmation with a phrase such as "Did you say [...]?" A possible problem with explicit confirmation may be that users tend to confirm such feedback too readily, without really paying attention to it (L.J.M. Mulder, Ph.D., personal communication, June 1997, Groningen University, the Netherlands). Another option is a backup facility. Whenever the user notices that something is misunderstood by the system, they can backup and provide the input again. Countermanding is another option, where the user can interrupt the discourse with a phrase such as "No, that's wrong", which will cause the system to interrupt the action it was about to take and do something else instead. The difference between explicit confirmation and countermanding is that the first option is initiated by the system—typically when a recognition has a low confidence measure—and the second option is initiated by the user—when they perceive an error.

The eventual choice for the type of error correction will be based upon considerations such as the type of application, the type of action the system is performing, efficiency, intrusiveness, cost and fatality of errors, and a trade-off between speed and accuracy.

2.2.5 Help

Another issue in the design of a VUI interface is how to provide help to the user without interrupting the flow of the interaction, and preferably only to users that require it. Since all input and output in a solely auditory interface can only be transferred in a sequence, help messages will slow down the interaction with an IVR system. This is acceptable as long as the help messages are needed and improve the interaction by preventing incorrect user input, for instance. When help messages, such as examples of possible input, are not needed by the user, they may become irritating, because the user has to wait for them to be played.

In an experimental study on the usefulness of a speech interface for a hand held audio messages recorder, Stifelman, Arons, Schmandt & Hulteen (1993) concluded that customization of both the amount and type of feedback—which is a type of help—is necessary in a speech interface. User initiated customization would slow down the interaction, however. An optimal interface would perform such customization automatically, based on the behavior of the user (Kamm, 1994), thereby increasing time-efficiency.

One possibility for providing help at appropriate times is to do this only when the user's speech is not recognized (on rejection), or when the user does not speak at all (on time-out). Besides providing help on rejection and time-out, a specific help command can be made available. In the latter case, the user needs to be instructed about that command, and reminded of it at intervals. How help is provided in a speech interface must be weighed along with the size of utterance-complexity, costs of errors, desired speed of the interaction, duration of system output, and so forth.

2.2.6 Turn-Taking

Human discourse contains numerous cues that provide guidance to the conversational partners on the dialogue itself. Among cues such as feedback on the status of information and the exchange thereof, these partially subliminal messages include

[4] The recognition accuracy level of 90% for in-set utterances can be much lower when users say out-of-set phrases.

indicators of who is the speaker and who is the listener, and at what moments these roles change. Indication of turn-taking is important for the efficiency of a dialogue, since it allows the speaker to provide all the information they deem necessary before the listener returns an appropriate response. When it becomes apparent to the listener that the speaker expects a response, the listener will evaluate the status of information exchange with the help of discourse indicators and become the new speaker.

These cues must be made apparent in the discourse between an ASR application and its user. Several possibilities are open here, since there are many different cues in human discourse that achieve this kind of indication. These indicators could include the wording and intonation of the prompts for example, or they may be a distinctive signal, like a beep tone.

When the recognizer does not have the 'barge-through' feature especially—which would allow users to interrupt the system output by talking before the end of the prompt, clear indicators for turn-taking are important for efficient interaction. Such an indicator could be a distinctive signal, like a short beep tone. Even when 'barge-through' is implemented, however, if users are unaware of the possibility of interrupting the system output, or they are used to systems that let them wait for a beep tone before they can speak—such as telephone answering machines, for example—they will wait for an indication of turn-taking. In these cases a more natural cue, such as a change in intonation, could be helpful.

2.2.7 Call-Flow Design

Flow-charts are a simple, but useful tool for dialogue design (Dix, Finlay, Abowd & Beale, 1993). Flow-charts that specify the course of the interaction between a telephone system and its callers are known as *call-flows*. The linear sequence of actions that is typical of dialogues is represented in a call-flow by the interconnection of system output and user input. This overview of the functionality of the system is abstractly represented by boxes and connectors.

As can be seen in Figure 2.4, in a call-flow the system's prompts are contained in rectangular boxes—mostly representing recognition states—and user input in parallelograms, while internal system actions and decisions are represented inside lozenges. Transitions between system output, user input, and system decisions are indicated by arrows. System output typically consists of a set of different prompts for each state: the opening prompt for the state in question and a time-out prompt. The time-out prompt is played when the system does not detect any input after the first prompt. Feedback is also included in the prompts, and can usually be found in the form of variable names in the prompt strings.

Figure 2.4 shows the call-flow for the train schedule menu as it would eventually be implemented in the Commuter Rail system as part of the experiment application for this study—see also section 4.4, *Materials and Apparatus*. In this state of the interaction the system reports the schedule for a certain train and then prompts the caller with further options. The user can request information on an earlier, later, or a different train altogether, or request for the information provided to be repeated. In the user input for this state, the word "train" and the phrase "this information" are optional. This means, for instance, that the system can recognize both "Earlier" and "Earlier train" as a request to search for an earlier connection. When the user reenters this state, either directly, or indirectly by specifying the data for a different train, the system will play a shorter version of the prompt, assuming that the user already has some idea of the possible options. This *prompt tapering* (Yankelovich, 1995 and 1996) is represented here by 'prompt1' to 'prompt3'. Please see pages 37-38 and 57-58 for more examples of call-flows.

Merely mimicking a DTMF interface may look like an easy and smart approach for building a speech interface, since the DTMF menu structure is already familiar and simple to implement. Such a speech interface, however, would not do justice to the flexibility, possibilities and naturalness of speech. Fay (1992; as described by Schumacher *et al.*, 1995) compared user preferences for DTMF and ASR input methods, demonstrating that it is useless to copy the DTMF menu structure in an ASR interface.

In a touch-tone interface the mapping of digits to menu options needs to be explained, and users are often instructed to end their input by pressing a designated button on the telephone key-pad. In a speech interface these lengthy and cumbersome instructions are not necessary, since commands can simply be called by their name. Speech also allows the combination of commands or data in a single phrase. Because natural language can often lead to a desired action more directly this way, the ASR interface structure will be very different from DTMF.

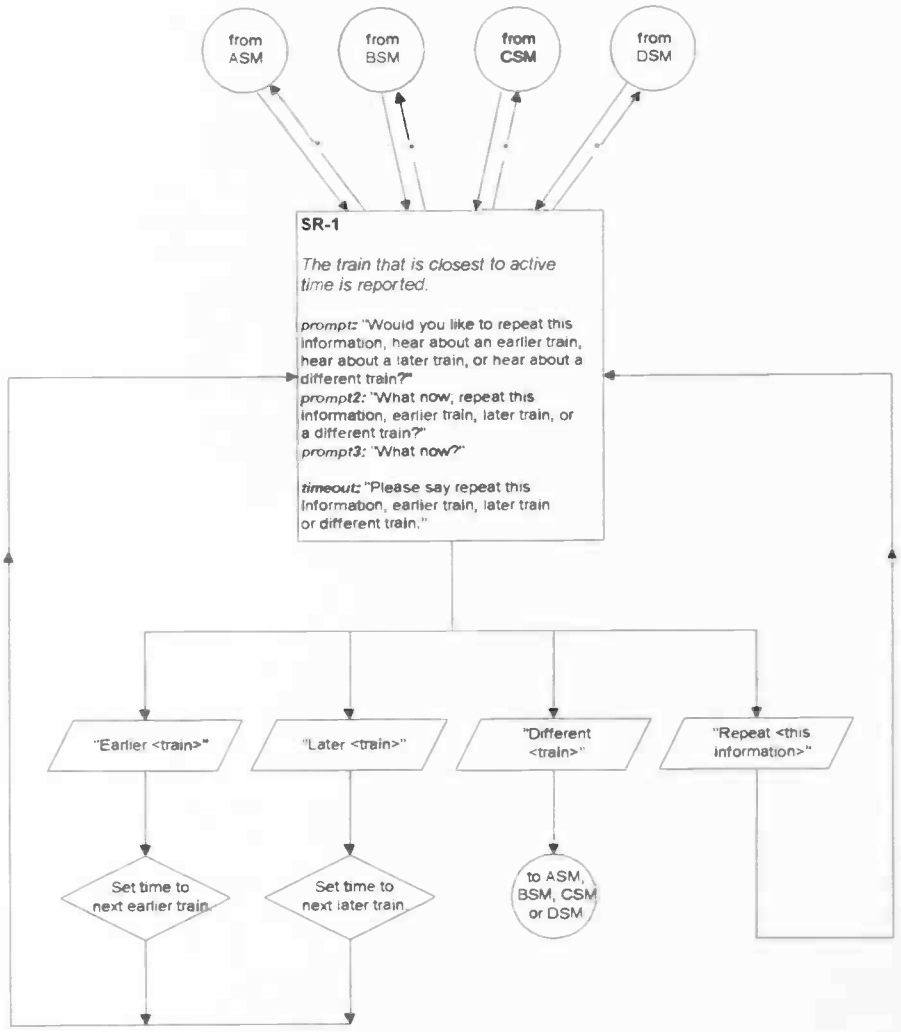


Figure 2.4—Call flow for train schedule report menu

2.3 Summary

By the use of DTMF interfaces, IVR systems have been deployed for years in various kinds of services. Since the desired functionality for these systems is running into the limitations of touch-tone interfaces, designers are turning to ASR as a possible solution. With advantages such as naturalness and high flexibility, speech can expand the boundaries imposed by touch-tone and improve user interfaces in their efficiency and usability.

The disadvantages of speech—possibility of misrecognitions and relatively high startup costs—can be overcome by the combined efforts of Human Factors and ASR Engineering. By taking certain VUI design issues into account, such as prompting, feedback, and turn-taking, Human Factors can provide an important contribution to enabling optimal use of the flexibility and naturalness that speech has to offer, while maintaining high levels of user performance.

3 Research Question and Rationale

A theoretical evaluation of the suitability of ASR for IVR systems, and of user performance and acceptance with respect to speech interfaces, leading to the proposal for the design of this study

IVR systems can only find their way into the commercial market if they are cost-effective. This means that the applications have to be able to handle a large number of calls in a short time, and satisfy every caller with their ability to be at service. If the system is not able to complete a session within a certain time frame or is not able to handle a certain number of calls at once, customers are not optimally able to reach the telephone information system to make use of its service. If the callers do reach the system but are not satisfied with how it works, they will avoid using it, and need expensive time with an operator or an employee at the company.

In this chapter a research question will be posed to examine the influence of factors that are part of the choices to be made in building a speech interface for automated telephone systems. The different approach towards interaction of DTMF and ASR—as illustrated in section 3.1—indicates that when the limitations of a touch-tone system impinge on the functionality of an automated telephone system, it may be time to consider replacing it with speech technology. Empirical support for the suitability of speech in this respect is provided in section 3.2. In addition to the technical performance of the system it is user performance and acceptance that have a direct influence on cost-effectiveness. Section 3.3 will elaborate on user performance and acceptance with respect to speech interfaces, showing that careful prompt design can lead to the optimal amount of user freedom and the necessary user input curtailment, as described in section 3.4. The research proposal is described in section 3.5, followed by section 3.6 on the expected results.

3.1 System Directiveness

Many automated telephone systems are very directive towards their users. They take the user step-by-step through their DTMF menu structure and tell the user what to do at each step. The user is told which key to press for which option, or to spell out words on the telephone key-pad. Here it is not the user, but the system that is in full control of the interaction. The following transcription of a call to the commuter rail schedule information line of the Massachusetts Bay Transportation Authority[5] (MBTA) will illustrate such a system-controlled interaction:

> MBTA: "If you are traveling on lines going towards or going away from North Station, press '1'. If you are traveling on lines going towards..."
> *Caller interrupts by pressing '1'.*
> MBTA: "For the Fitchburg line, press '1'. For the Lowell line, press '2'. For the Haverhill-Reading line, press '3'. For the Rockport-Ipswich line, press '4'. To return to the main..."
> *Caller interrupts by pressing '4'.*

[5] The MBTA telephone information center can be reached at +1-617-222-3200.

MBTA: "For information on the last train from North Station, press '1'. If you are traveling on lines *to* North Station, press '2'. If you are *departing* from North Station..."
Caller interrupts by pressing '2'.
MBTA: "Please enter the first four letters of the station you are departing from."
Caller presses '7625', where the digits correspond respectively to "ROCK" on the telephone key-pad.
MBTA: "Rockport Station. If this is the correct station, press '1' *now* to continue."
Caller presses '1'.
MBTA: "For departures from Monday through Friday, press '1' *now*."
Caller presses '1'.
MBTA: "All time entries must be concluded by pressing the pound key. Please enter the time you wish to arrive, followed by an 'A' for AM, or 'P' for PM. Conclude your entry with the pound key. For example, for three o'clock PM, you would enter '300P', followed by the pound key. Please enter the time you wish to arrive, *now*."
Caller presses '0730' and waits during the period of silence that follows.
MBTA: "I'm sorry, you've entered the time incorrectly. Please reenter your arrival time *now*."
Caller presses '0730' and waits during the period of silence that follows.
MBTA: "Please hold while I transfer you to an operator. You'll hear a click followed by a brief period of silence."

The speech recognition community increasingly takes a very different approach. Far from directing the user's every move, it is actually continually pushing recognition in the direction of allowing users to say *anything*, thereby giving them complete control over the discourse. An example of how such a system functions in real life can be found in Europe, where the Philips Research Laboratories in Aachen, Germany developed a train schedule information system (Aust, Oerder, Seide & Steinbiss, 1994). The Telefonische Automatische Bahnfahrplan-Auskunft (TABA) welcomes its callers with the phrase "Wie kan ich Ihnen hilfen?" ("How may I help you?"), as in the following translated transcription (Aust *et al.*, 1994):

TABA: "Good morning. This is an automatic timetable information service. How may I help you?"
Caller: "I need a connection from Hamburg to Munich."
TABA: "When do you wish to travel from Homburg to Munich?"
Caller: "No, *Hamburg* to Munich."

Which approach is suitable for what type of application—whether the system-driven or the free approach—is by far not clear-cut. In fact, it is not even clear whether one extreme would have a distinct advantage over the other. What leads to a high efficiency in one application may be inefficient in another. The same counts for user comfort.

It is evident that a system-driven DTMF menu structure is too inflexible and unnatural for the desired functionality of contemporary IVR systems. Intuitively, one would say that an open question in a speech interface, allowing the user to formulate their question or command in any way they desire, permits them to communicate in the most natural way with the system. Interacting with a system is not the same as interacting with another human being, however. Users have a different attitude towards interacting with machinery compared to interacting with another person. It may also take more cognitive effort on the part of the user to respond to a general query than to a specific one. When open-ended prompting is incompatible with users' attitudes towards an automated system, or when the necessary cognitive effort is uncomfortably high, allowing users to say anything may be counter-productive.

For any specific application it is necessary to investigate what amount of user freedom works best, especially when the technology used allows for such a degree of freedom that it begins to pose problems to the interaction itself.

3.2 Suitability of ASR for Telephone Information Systems

One of the promising benefits of input in speech form is that it adds an extra modality—namely, sound—to tasks that require keyboard, or other kinds of input, allowing user attention to be divided more efficiently over different modalities. Speech offers another mode of communication that may be used in some contexts to supplement existing channels in a user interface (Dix *et al.*, 1993). Such an existing channel would typically be vision, as in a Windows environment, for instance. Although telephone information systems cannot benefit from the visual advantage, since their interfaces are solely auditory, the advantages of speech—reviewed in Chapter 2, *Speech Interfaces for Telephone Systems*—nevertheless make ASR an excellent candidate for IVR interfaces.

When speech is to be chosen as the modality for an IVR interface, it is especially important to find ways of handling the many problems that emerge when current technological limitations are combined with the complexity of speech[6]. Since all the information in an interaction with an IVR system will be transferred over the same auditory modality, proper interface design is essential in providing a clear distinction between input, output, and feedback. Since parallel channels are lacking in IVR system interfaces, all transactions between the system and the user occur in a sequential manner.

The principal advantages of speech that are relevant to IVR systems, as we have seen in Chapter 2, are its speed and flexibility, which translate into interface efficiency and user comfort. It is this efficiency and user acceptance that are the main points of interest of the experiment described in this report. The actual choice for ASR will depend on whether it can be implemented into a system that is both efficient and user friendly. The empirical support for the applicability of speech in such a system is outlined in the next paragraph.

3.2.1 Empirical Support for Usability of Speech

Although the availability of research in this area is still low, existing empirical support indicates the usability of speech as a modality for interfaces of automated systems. As Martin (1989) suggests, from an intuitive standpoint human-computer interaction should be enhanced by the speed of speech. Typing and especially spelling out words on a telephone key-pad—a tedious procedure that is currently much in use with telephone information systems—is significantly slower than speech, as we have seen in section 2.1.2, *Level of Abstraction*. With respect to IVR systems, spelling out words on the telephone key-pad is even slower than typing on an ordinary keyboard. Martin (1989) extends evidence for the claim that speech is faster than typing with an empirical study in which subjects performed a highly interactive computer chip design task, by giving commands with mouse clicks, single key presses, and typing or saying full words. Subjects performed four two-hour sessions with a design package they were familiar with, but had speech recognition added to it. In half of these sessions they were encouraged to use speech, while for the two control sessions speech was not permitted. For these design tasks, speech turned out to be a more efficient response channel than typing. In this case the subjects used an ordinary keyboard. With a touch-tone IVR system, however, callers would have to use the telephone key-pad for spelling out.

Research on the efficiency of speech as a modality for an interface produced contradictory results as well. While Poock (1980, 1982; as described by Damper & Wood, 1995) claimed that speech input was faster than typing and also less likely to suffer from errors, Damper *et al.* (1995) showed that speech was somewhat slower and considerably more error-prone than typing. They did this with an experimental design

[6] Please see paragraph 2.2, *VUI Design Issues*.

that was set up as a critical analysis of Poock's results. Their reproach was that Poock had used commands that were unreasonably long for the keying condition of his experiment, in which subjects had to enter naval commands and control instructions onto a military network. By using terse commands for the keying condition instead—for instance "LRC" instead of "Launch Rescue Craft"—and the full commands for the speech condition, they showed that Poock's approach had introduced a bias in favor of speech recognition.

The contradictory empirical findings mentioned above indicate that the advantage of speech over typing also depends on the way ASR is deployed. For IVR systems speech is clearly a more efficient response channel than the telephone key-pad, especially for specifying names. When the advantage of speech differs for different implementations, it may also be the domain of the application that affects the usability of speech.

That speech is especially suitable for certain kinds of interaction has been assessed experimentally by Martin (1989), who evaluated a study by Nye (1982). In this study, where subjects performed a baggage handling task, speech was only faster than keying when users could say destination city names instead of the three-digit codes that were used for it with keyboard input. This corresponds to the subsequent Damper *et al.* (1995) findings described above. Martin (1989) concludes that speech as an input capability allows for a change in the nature of the input. This means that speech may not always be faster, but that a redesign of the task can maximize its advantage where it shows superiority over DTMF. This may be an argument for not just replacing touch-tone structures with an identical speech version.

By comparing different speech input simulations, Martin (1989) also confirms that speech has a high response efficiency. High response efficiency allows the user to respond quickly. This will lead to a significant improvement of time-efficiency only when the task in question does not involve a significant amount of time-consuming cognitive processes spent on parts of the task that are not directly related to the interaction. When not the interaction itself, but other processes, such as document composition, for instance, take up most of the time of the task, high response efficiency may not contribute significantly to better time-efficiency. Martin (1989) states that high response efficiency may be most beneficial for "tasks of a high[ly] interactive nature", that "do not require considerable 'thinking' time between transactions."

A telephone information system, which the experiment described in this report is concerned with, is one such domain of a highly interactive nature. Within a relatively short time frame such a system will ask the user a series of questions which will provide the necessary pieces of data for the system to conduct a search query or formula. The user is able to provide these pieces of information without having to spend considerable time thinking about them. In the interaction with a telephone system, there is no other ongoing activity but the discourse between the system and the user. After the user has given one or more pieces of information, the system will immediately ask for further data, until it has enough to be able to provide the requested information. A call to the MBTA telephone information center, transcribed on page 16, is such an interaction. This kind of interaction is typical for what Martin (1989) considers speech to be a very suitable modality because of its high response efficiency. With a speech interface, the MBTA interaction described earlier could look like this:

1. question-answering task
2. restricted system domain
3. users occasional
4. users with good knowledge of the system domain
5. ad hoc system questions
6. immediate users answers

Table 3.1—Characteristics of tasks for which natural language is most likely to be effective (Capindale & Crawford, 1990)

MBTA: "From which statin *to* which station would you like to travel?"
Caller: "From Rockport to North Station."
MBTA: "From Rockport to North Station. Around what time and on what day of the week?"
Caller: "Around 7:30, on Thursday."

Additional support for the claim that speech is suitable for telephone information systems comes from an extensive review by Capindale *et al.* (1990) of a variety of experimental work on the usability of speech recognition in different applications. They conclude that natural language is indeed likely to be the most effective method of interaction for certain specific tasks. These tasks have a number of features that are characteristic for interactions with telephone information systems, such as a transit information system, for instance. These tasks typically involve immediately answering ad-hoc questions from the system by occasional users that have good knowledge of the restricted domain—see also Table 3.1.

Also, most users of IVR systems are likely to be occasional, since they only use the system when they need the particular service or information it provides. Occasional use of a system inhibits command language training, but with a speech interface users with good knowledge of the system domain are able to quickly answer the questions the system asks, as long as sufficient instructions are provided about language restrictions.

3.2.2 Conclusion

It is evident that speech recognition has much potential. When it is employed in user interfaces, it can vastly improve the efficiency of the interaction with an IVR system. When the interface is properly designed it will also lead to a high level of user friendliness and hence have a high level of user acceptance. Some contradictory empirical findings, however, indicate that designing a good VUI is not simple.

Careful speech interface design is necessary to maintain a clear distinction between the different types of information in the dialogue, such as instructions, requests for input, and feedback, since they travel over the same auditory channel. Empirical findings supporting the usability of ASR in automated systems show the high response efficiency of speech, which makes speech interaction generally faster than interaction by keyboard or the telephone key-pad. The higher response efficiency of speech can only be exploited by careful implementation, however. Also, as we have seen on page 19 of this section, the domain of the application has an influence on the advantage of speech. For the highly interactive question-answering tasks common with IVR systems natural language is a very effective method of interaction.

The capabilities and limitations of the speech technology used will define a starting point for Human Factors considerations of the design of the interface. By taking cognitive psychological and perhaps sociolinguistic considerations into account, an efficient and user friendly ASR application can be built despite the limitations of the speech technology. An empirical clarification of the aspects of user comfort and efficiency is necessary to allow Human Factors to develop designs that optimally exploit the potential of ASR.

3.3 User Performance and Acceptance

Whether a directive or more non-directive type of interface is best for IVR systems and whether only simple or also more complex speech input is to be preferred depends on a number of different aspects. Irrespective of the modality of the interface, these aspects will always include the capabilities of the technology used and of the system itself, attitudes of the users to automated systems in general, including psychological factors,

such as the degree of confidence the user has in the capabilities of the system, or the perceived amount of control over the interaction.

The status and influence of some of these aspects are well known already. The capabilities and limitations of the technology that is being used, for instance, will always be known from testing and experiences with previous implementations, and serve as a clear starting-point for the design of an application.

The specifications of the technology used are not the only factor, however, that has an influence on determiners of system quality, such as time-efficiency, task-efficiency, and task compliance. Factors concerning users' attitudes towards the system are also to be considered. Some of these factors, such as the user's perception of control, are not as clear as the specifications of the technology, but can be supervised. Yet other factors, such as the degree of user confidence, need to be studied more closely before they can be controlled or at least influenced in a useful manner, since they are interconnected with a number of other aspects in a complicated way.

3.3.1 User Comfort

We are becoming increasingly accustomed to interacting with machinery and systems, and the need and usefulness of IVR systems is now widely acknowledged. There are, nevertheless, still some users who prefer not to interact with automated information systems at all, but insist on talking to a human operator instead. But even people who do use contemporary IVR systems, also show dissatisfaction, and frustration even, with the design of these systems (Schumacher *et al.*, 1995). This means that much still needs to be done to improve user comfort and acceptance of automated telephone systems.

Since speech is people's most natural way of communication, its use in an interface may be expected to lead to a higher degree of user comfort. Certain issues need to be taken into account, however, before this expectation can be realistic. With ASR systems, there are three major factors that have an effect on user comfort.

The first factor is the ability of the system to recognize the user input. If its formatting is completely left to the user, it is harder for the system to predict what it might receive as input. The system has to be able to allow a wide range of possibilities, but with larger recognition contexts, recognition will be harder. Although speech technology is vastly improving, the combination of speaker independent, phone-band[7], continuous speech limits the size of recognition contexts[8] considerably when high levels of recognition accuracy need to be maintained. This limitation has a direct impact on the freedom of the user. Much can be done with keyword or phrase spotting, but the task will become considerably harder when meaning has to be deducted from the often ambiguous sentence structures that are inherent to natural language. Difficult recognition may mean that the system just needs more time to perform, but it may also mean that it makes more errors, or has a higher chance of not being able to recognize the input at all. From the user's point of view a system that is either slow, or inaccurate, or both, is not comfortable to use.

---

[7] The term 'phone-band' refers to the limited frequency range of an ordinary telephone line. Karis & Dobroth (1991) specify this range as 300 to 3400 Hz. 'Wide-band' refers to the frequency range of speech input that enters through a microphone directly hooked up to the system. On personal computers, speech is usually sampled at 22,050 samples per second, which means that the maximum frequency that can be detected is 11,025 Hz. On other systems it is very common to use a 16,000 Hz. sample rate, leading to an upper bound of 8,000 Hz. The lower bound of the 'wide-band' frequency range will depend on the threshold-frequency that is chosen to reduce noise—this could be 130 Hz., for instance.

[8] Please see section 2.2.2 for an explanation on *recognition context*.

Besides these limitations on the side of the system, there is a second factor that may influence user comfort. The large degree of freedom that speech allows may induce discomfort in users. In addition to being accustomed to receiving very precise instructions from automated telephone systems that work with DTMF, users may be uncertain about how to formulate their input. They have never had the freedom of natural speech in previous systems they have known, so they just do not know what to do with it once they have it.

A third factor of possible influence on comfort is the users' confidence in the system's capabilities of recognizing and understanding their speech input. Especially when users realize that the recognition is faulty, discomfort may be induced by a fear that the system may not understand what is being said. A dialogue can be properly maintained only when both conversational partners know that the other party is able to listen and understands the context of the conversation. Therefore, a limited context must be defined for the interaction and made clear to the user.

All these issues are important for the way an ASR application interacts with its users. The larger the discrepancies between the approach of the system and the approach of its users, the greater the likelihood of lower efficiency and user comfort.

3.3.1.1 Sociolinguistic Considerations

With powerful ASR applications that are being built under the assumption that people prefer to communicate using unrestricted speech, users are not required to learn how to phrase their input. However, there are reasons to suspect that users might be equally or more satisfied with a more controlled interaction:

1—*Automated systems and machines are widely used for obtaining information or services, and people are accustomed to system-driven interactions.* Besides the wide variety of IVR systems, people use machines such as Automated Teller Machines (ATM), train ticket machines, parking ticket machines, etc. The typical interaction with these systems is completely directed by the system instead of by the user, and is generally efficient in terms of error rate, task compliance and time-efficiency. If people associate restricted interactions with the efficiency of machines, owing to their past experience with automated systems, then there may not be a special need for more free interaction. It might even make users doubt the reliability or efficiency of ASR applications that do not use such restricted communication. Buchheit & Moher (1990) conducted an experiment in which subjects had to fill out a multiple choice questionnaire, to determine expectations in human-computer discourse. The results supported their intuition that people would expect a consistent and higher level of assertiveness from computers than from human beings. Zoltan-Ford (1989) found that subjects did not feel hampered by imposed restrictions on their speech input, because they expected computers to have a limited vocabulary and a need for at least some consistency, and they thought that communicating with a computer required prior learning. With automated systems, people might feel uncomfortable about being in control, and prefer a more directive system for that reason. The image of the computer as a cold, methodical device may be less threatening than a model that is more human-like (Buchheit *et al.*, 1990).

2—*System-driven interactions are free from the costs of social interaction.* In ordinary human-to-human conversation, a certain amount of time is spent on social formalities such as greeting the conversational partner. The amount of time spent on the social part of the interaction will depend on various aspects, such as culture, for instance, but in an efficient system-driven interaction time will probably not need to be spent on this. Since social formalities and politeness are inherent to inter-human communication, people may use or expect, if only subconsciously, the social part of the interaction

with natural language in ASR applications as well. Although users will not expect the machine to say "How are you today?"—since a high level of socialization is irrelevant for the mere task of interacting with an automated information system— there is a need for a certain degree of politeness. Systems represent the company that provides the users with the service, so what the system says is really the company talking to its clients.

3—*People may feel uncomfortable in situations in which they do not know how to formulate what they wish to say.* People are used to the need for formatting their input in a certain way when interacting with automated systems. They realize that this is necessary for the system to be able to handle their input. Subjects of the Zoltan-Ford (1989) study on shaping and modeling user input stated in post-session interviews that they were not hindered by restrictions, because they did not expect the computer to have the same vocabulary as its users. Similarly, when interacting with an ASR application, users are likely to assume that not everything can be recognized either. This assumption is quite correct, since even the most powerful speech recognizers restrict to some extent the domains and format of the input. If such systems give or seem to give their users complete freedom over how to formulate their input, the users may feel uncomfortable about not having prior instruction on how to do that best.

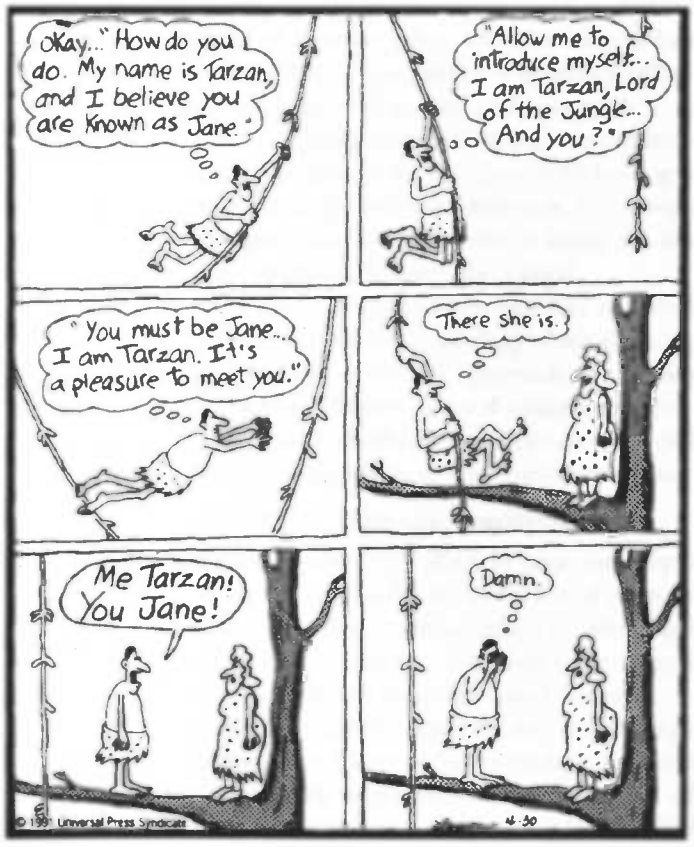


Figure 3.1—"People may feel uncomfortable in situations in which they do not know how to formulate what they wish to say."

4—*Individual differences exist in the amount of information that people tend to give in a single utterance.* In natural language, some people may consistently use short discrete sentences instead of more complex ones containing a large amount of information, while other people may prefer lengthier phrases. If complete user freedom were allowed in the interaction with a speech application, so that users can either say complex or short phrases, this would not pose a problem. In practice, however, a certain amount of predictability of the complexity and formatting of the user input is necessary for the recognizer to handle the speech input. To achieve this predictability, the application will attempt to induce its users to phrase their input in a certain limited way. Users may be given a sense of more freedom and the interaction may appear to be more natural when the input is limited to more complex phrases as opposed to short speech commands. Unfortunately, it seems to be more difficult to induce people to say longer utterances of a certain format than to make them to say specifically formatted short utterances (Zoltan-Ford, 1989).

3.3.2 Efficiency

ASR applications can exploit the communication skills and expectations that users have developed through everyday experiences. A more efficient and effective transfer of information between human and machine can thus be created (Leiser, 1989; as referred to by Kamm, 1994).

Making use of aspects of human dialogue does not in itself automatically lead to a faster interface. In a speech interface additional time will be needed, although this can not be too much, for the necessary feedback. Also, the phrasing and processing of the speech input will cost a certain amount of time.

Although newer versions of speech recognizers almost without exception demand more processing power from the computer they are running on, they are also that much more capable of accomplishing a higher recognition accuracy. In addition to this higher level of accuracy, they are often able to handle larger recognition contexts. The ASR community is continuously investing effort in improving the capability of their technology to deal with complex speech.

A more powerful speech recognizer is not in itself a guarantee of better cost-effectiveness, however. Speech technology is improving, but still has limitations that make a continued investment in VUI design necessary. The application needs to be designed in such a way that it limits the interaction to a certain domain, and prevents the user input from being too varied.

A perhaps less obvious but nonetheless important reason why the ongoing improvements in technology alone do not guarantee higher efficiency and better user acceptance, is that the approach of allowing users to say *anything* in itself just may not be the most efficient. Even if complex speech can be recognized, complete user freedom may not be most efficient because of sociolinguistic reasons—outlined above—and certain technical reasons—outlined below.

3.3.2.1 Technical Considerations

Whether allowing callers to state their input using complex speech makes an application more efficient is by far not a trivial issue to resolve. On the one hand, complex speech speeds up the communication by allowing multiple pieces of information to be specified in a single utterance, therewith decreasing the number of necessary steps in the interaction. On the other hand, however, it slows down the interaction by the need for extra time for the phrasing and processing of the speech input of longer duration, in addition to the need for extra time to play lengthy prompts that instruct users how to phrase their input and request the pieces of data needed. It is evident that the higher response efficiency of speech (Martin, 1989) does not automatically lead to an increased time-efficiency. By weighing the principal technical considerations, an answer will come closer within reach.

1—*Complex utterances may lead to faster interaction, because fewer steps are necessary to complete it.* When complex input is permitted, the interface allows the user to specify their query or command directly and completely in a single or just a few steps, thereby bypassing sequences of input iterations. Decreasing the number of necessary steps can produce a dramatic increase in time-efficiency. For every step that is not necessary, time is saved on system output, waiting for user input, processing user input, etc.

2—*Paradoxically, complex utterances may also lead to slower interaction, because of the time spent on pronouncing extraneous speech and prompting for complex input.* In comparison with a very straightforward system-driven interaction, natural speech contains extra information that is not useful to the system. The user will spend an amount of time to provide this extraneous speech which adds to the total duration of the interaction. When

more complex utterances are used, lengthier system output is necessary to prompt the user for the several pieces of data that are expected at each step. In a drive to minimize the time spent on every incoming call, several directory assistance services in the United States have eliminated the natural aspect of the conversation between operator and customer, thus paying the price for efficiency. They use recorded messages such as "City please" and "Which listing?" to prompt the caller for input, while a human operator listens and acts upon what is being said without wasting time on social formalities. Not surprisingly, many people are under the mistaken impression that automatic speech recognition is being used for these services.

3—*More processing power is needed to recognize complex phrases.* The higher processing cost of recognizing complex speech has two consequences. First, to prevent computational complexity from slowing down the interaction, faster and more expensive hardware may be needed. Second, to prevent computational complexity from limiting the number of concurrent calls that can be handled, still more hardware may be needed. If the need for extra computational power to recognize complex speech exceeds a certain threshold, the cost-efficiency of a speech interface will be lower.

4—*The user needs more time to think of how to phrase complex utterances when they are required in a given format.* Depending on the domain of the information system, users may need more time to construct their input—before they actually say it—when the system expects their phrase to contain several pieces of information. Certain pieces of information may simply be less intuitive to put together in one sentence, or they may be more difficult to put together because they are all complex figures, for example. In addition, when total user freedom is allowed, callers may need more time to construct their input for the lack of examples after which to model it.

5—*Increased user freedom multiplies the chances of misrecognition.* No matter what the power of a speech recognizer may be, complex speech will always be more difficult to recognize than simple phrases. Complex speech introduces larger recognition contexts, and extraneous speech can only be filtered out by rejection—*i.e.* by not being recognized as something useful, which means that is has to be processed as well. The process of computing a recognition over a complex sentence in a larger recognition context increases the possibility of misrecognition.

3.3.3 Conclusion

A difference exists between the question of the amount of user freedom for ASR that is able to recognize everything, and for ASR that has certain limitations. In the first case the problem is whether a large degree of user freedom is efficient or not and whether users feel comfortable with being in control of the interaction. In the second case, there is an additional problem, which is how to induce users to phrase their complex input in the required format while making the system appear to have an interaction that is natural, as if the user indeed does have a lot of freedom.

Since contemporary speech technology is not able to recognize *everything*, user input needs to be controlled. With a higher input predictability, recognition contexts can be smaller and consequently recognition accuracy improved. Better ASR performance will increase user acceptance and overall efficiency, thereby making speech a more cost-effective solution for the modality of IVR interfaces.

That the user input has to be controlled does not necessarily mean that input phrases have to be terse. As long as the system is programmed to have reasonable expectations of user's responses, it should be able to handle input that is more complex than single words. But the grammar should not be too complex either. While total user

freedom may induce discomfort in the user because of an uncertainty of how to formulate the input, it will also hamper recognition because of the lack of predictability on the side of the system. Ideally, the system would induce input of a complexity that is just high enough to appear natural to the user, but still simple enough for the system to predict and recognize. A system that appears natural and elicits intuitive responses will leave its users with a high degree of comfort. Thus, the choice of utterance complexity can be an important one with respect to user acceptance and both user and system performance.

3.4 Prompting and User Input Curtailment

Even the commonest of human discourse contains a complexity most people are not aware of. Bunt (1988; as described by Engel *et al.*, 1993) found, for example, that in inter-human communication more than half of the utterances are secondary messages. Secondary messages do not contain information directly related to the topic of the conversation. Instead they are clues containing information about the dialogue itself. The clues that are included in communication between people indicate discourse control issues such as turn-taking, whether something is being asked for, or given information on, or to refer to what is understood, and to refer to what is expected.

A field study by Zoltan-Ford (1991) provides a clear indication of a way to control user input. By letting subjects interact with an inventory program, in a 'Wizard-of-Oz'[9] type of setup, she tested whether users would automatically model their input after the consistent vocabulary and grammatical structure of the program's output (modeling). She also tested whether they would rephrase their input according to the program's style of communication when the system did not understand input of a different style (shaping). The following two transcripts of an interaction that could be typical for a loan calculator application will illustrate the difference between modeling and shaping:

MODELING:

System: "Over how many years, at what interest rate?"
User: "Over 10 years, at an interest rate of 8.5%"

SHAPING:

System: "Over how many years, at what interest rate?"
User: "The loan will be paid over a period of 10 years, and the interest will be 8.5%"
System: "I'm sorry, I'm not sure what you said. Over how many years, at what interest rate?"
User: "Over 10 years, at an interest rate of 8.5%"

When modeling does not occur, the necessary curtailment can be achieved with shaping after all, according to a more explicit template. Two findings of the Zoltan-Ford (1991) study directly indicated the possibility of reducing the variability in user input by modeling and shaping:

1—*For the sets of vocabulary tested, user input could be modeled and shaped after the communication style of the system regardless of the vocabulary or length of system output.* Mean length of user input in the experiment was influenced by the length of computer output. The modeling was not perfect, but the influence was significant, indicating that different prompting techniques elicit different responses from the users.

[9] In a 'Wizard-of-Oz' setup, subjects are told that they are interacting with an automatic system. In the case of an ASR study, subjects would talk to an application ostensibly capable of recognizing their speech input while an experimenter is actually monitoring what the subjects are saying. Depending on the subject's input, the experimenter triggers predefined responses of the system. The system output that is then played appears to the user as the system's reaction.

2—*The variability of user input was better reduced by shaping it than by relying on the users to model their input.* The effect being stronger in the shaping condition than in the modeling condition indicates that curtailing user input can be improved when the user is made aware of the limitations of the recognizer.

Zoltan-Ford's (1991) experiment concludes that the design of the prompting is of considerable importance for speech interfaces. By using the same conversational style in the computer output as is required for user input, the input can be made more predictable. User comfort can be increased by providing either implicit examples—through the use of a certain vocabulary and grammar—or explicit examples. Explicit examples will be more closely related to shaping and will therefore be desirable especially in cases where it is difficult for users to model their input after the system output.

As the Zoltan-Ford findings have shown, modeling and shaping indicate that different prompting techniques may elicit different responses from the user. This means that a certain level of utterance complexity in the user input can be achieved by using a corresponding level of utterance complexity in the system output. Different prompting techniques will therefore determine the appearance of the system, the speed of the interaction, and the level of directiveness. All these factors determine the efficiency of the interaction and the comfort of the user.

3.5 Purpose of Experiment

The primary purpose of this experiment was to examine the effects of different prompting methods on user acceptance, user performance, and task compliance. These three factors are the primary factors of importance with respect to cost-efficiency and user comfort.

Automated telephone information systems suffer from a set of inherent limitations that prohibit the use of conversational speech. While the frequency range of the public telephone network is probably the most obvious restraint, other qualifying factors are time pressure and the fact that most users will be novices with respect to the use of the system. The average users' lack of knowledge of what to expect makes it imperative that the interface be straightforward and easy and intuitive to use. At the same time, the need for time-efficiency means that it is not possible to provide extensive explanations on how to work with the system. When more time is needed per interaction step, or a larger number of steps is necessary, time-efficiency will be lower.

In order for telephone speech application interfaces to be efficient and user friendly, they should make the interaction appear as natural as possible, by invoking the correct, but natural response from the user. The need for an interface that induces users to modify their behavior to fit the requirements of the technology calls for an evaluation of the choice of utterance complexity, examples of user input, and conversational cues. This study investigated the combined effects of simple versus complex prompts, with the absence versus presence of input examples, with the absence or presence of beep tones, on user performance and acceptance.

3.5.1 Utterance complexity

As outlined in section 3.4, *Prompting and User Input Curtailment*, the phrasing of user input is influenced by the wording of the system prompts. Complex-utterance prompts, for instance, are likely to elicit complex responses. In this experiment, different prompting techniques were used to examine the effects of complex versus simple utterances on user preference and task completion time.

Subjects provided a test system with a predetermined number of pieces of data, in order to obtain some information. The utterance complexity of the interaction was varied

by either prompting the subject for a single piece of information at a time, or by prompting the subject for more pieces of information at each step. Subjects were requested for a single piece of information with a simple prompt, and for more pieces of information with a more complex prompt. For all experiment conditions, the system needed the same amount of data. Therefore, the number of interaction steps differed for the simple versus complex conditions. More information was provided with the complex utterances, thus making it likely that the complex condition would be more time-efficient by a smaller number of steps. Complex utterances, however, increased total task completion time by the time needed for the longer input and output phrases. Utterance complexity was also expected to have a certain influence on the naturalness of the system. More conversational speech would be more natural, perhaps leading to higher user comfort. On the other hand, complex utterances were expected to be more difficult to formulate than simple utterances.

3.5.2 Examples

The effects of the presence or absence of input examples were also analyzed. While both the simple-utterance and complex-utterance prompts were expected to serve as an implicit template for the user input, examples added to the prompt would act as an additional explicit template. Intuitively, an example would make it easier for the user to formulate complex input. On the other hand, the provision of examples slowed down the interaction because of the time needed for the examples to be played to the users.

3.5.3 Beep tones

The experiment also studied the effects of beep tones at the end of the prompts. For systems that are not equipped with the 'barge-through' feature, a beep tone may be a necessary indication of turn-taking. When 'barge-through' is not available, an indication of turn-taking would always need to be inserted at the end of the system prompts that request the user for input. For such systems, beep tones are necessary in two cases: first, when the prompt endings are not easy to detect, and second, when users have a tendency to interrupt the prompts by starting to talk before the end of the prompt. The recognizer that was used for the study did not have the 'barge-through' feature. The inclusion of the beep tone treatment was expected to provide insights into the usefulness of conversational cues.

3.6 Expected Results

3.6.1 Simple versus Complex Utterances

In accordance with Zoltan-Ford's (1991) findings, it was expected that users would adopt the conversational style of the experimental application. In the simple-utterance condition users would give short phrases as input, and in the complex-utterance condition the input would consist of more complex phrases containing more information and also more extraneous speech. Table 3.2 compares an example complex-utterance interaction and a simple-utterance interaction to illustrate the difference between the complex and simple condition.

Because adopting a simple interaction style would be easier (Zoltan-Ford, 1991), it was expected that the simple condition would be more comfortable for the subjects. This would be especially true for novice users (Capindale *et al.*, 1990). All subjects for this experiment were novice users, as they did not receive any training prior to the experiment. This was an important condition, since an IVR system such as a Transit Information System should be able to be at service for novice users especially. Such a system has to be able to help all callers, even when they call in for the very first time, and

simple utterance interaction:

1 *System:* "Which station would you like to leave from?"
Subject: "From Harvard Square"

2 *System:* "Harvard Square. Which station would you like to arrive at?"
Subject: "Park Street"

3 *System:* "Park Street. Around what time?"
Subject: "Around 8:30 AM"

4 *System:* "8:30 AM. On what day of the week?"
Subject: "On Friday"

complex utterance interaction:

1 *System:* "From which station to which station would you like to travel?"
Subject: "From Harvard Square to Park Street"

2 *System:* "From Harvard Square to Park Street. Around what time, and on what day of the week?"
Subject: "Around 8:30 AM on Friday"

Table 3.2—Example transcription of a simple utterance and a complex utterance interaction with a subway schedule information system

it is likely that most users would not call in to the system on a regular basis and become experts by frequent use.

This higher level of user comfort would yield higher user acceptance for the simple condition. Another reason why the simple version may have been preferred over the complex version is that step-by-step prompting would have presented the system as an efficient and straightforward methodological piece of machinery, inspiring confidence in the user. From field trials with two information inquiry systems Billi, Castagneri & Danieli (1996) concluded that people associate a speech telephone system with a traditional automated system when the interaction uses a simple grammar and is system driven, and with human operator services when the speech used is more complex.

Higher preference and performance measures for complex utterances would arise when their use was significantly faster than simple prompting. Complex prompting would be more likely to be faster when users did not have to spend time on thinking about how to phrase their response. This would be the case when it was more intuitive to combine the necessary pieces of information in a single phrase. It would be less straightforward to ask for every piece of information separately in cases where they combine naturally. On the other hand, complex utterances would also be preferred if the complex condition was sufficiently simple. Since the complex condition was only somewhat more complex than the simple condition, but still did not consist of conversational speech, it was expected that the complex condition could also have a satisfactory user comfort rating, or even be preferred over the simple condition.

The simple system was generally expected to be more efficient (Capindale *et al.*, 1990). User performance in terms of total task completion time would depend on the trade-off between step-size and the number of steps necessary for the interaction. As described above, performance would depend on the intuitiveness of data combination in single phrases. Recognition accuracy by the system was expected to be lower for complex utterances, thus lowering performance rates. Performance measures therefore depended on the provision of input examples, as described below.

3.6.2 No Examples versus Examples

In the Zoltan-Ford (1991) study, subjects quickly changed their inputs to resemble the program's output when recognition was restricted to modeled input only. Otherwise they

modeled their input to a much lesser degree. This implies that any restriction on the input should be made clear to the user, either by proper feedback, or by instruction. Restriction on user input will always exist, because of the inherent limitations of any recognizer—in fact, the subjects for our experiment were always restricted to either simple or complex utterances.

For the example condition of the study it was expected that examples would act as instruction on how to phrase the input, thereby making input modeling easier for the user. Where the input modeling was relatively difficult, in the complex condition, examples would be especially useful. For these conditions, user performance was expected to improve from the provision of examples. In the simple condition, where modeling was relatively easy, examples might be unnecessary—thus not leading to performance improvements—or even unpleasant.

It was also expected that different people would react very differently to examples. With respect to the simple-utterance condition especially it was expected that some subjects would have a negative attitude towards examples because these might be experienced as superfluous in this condition. Depending on the strength and variability in attitudes towards 'unnecessary help', it was expected that examples could have a negative effect on user acceptance for the simple-utterance condition

3.6.3 No Beep Tones versus Beep Tones

A beep tone at the end of a prompt would be a clear indication to the user that they could start talking. Such a specific indication would not be necessary when the prompt itself provided enough clues about turn-taking. Longer prompts were more likely to elicit a need for indicators of turn-taking, because with longer prompts it would be harder to know when the prompt finishes. Therefore, beep tones were expected to be especially useful in the complex-utterance conditions and in the conditions with input examples. The prompts of the simple-utterance condition—especially those without examples—are short single phrases with a clear start and ending.

Also, when prompts are short the user does not need so much to start talking during that prompt. Long explanatory prompts are much more likely to be interrupted when the user thinks that it is clear what is being asked for. Examples especially might be interrupted when the user does not feel the need to hear them out, and even more so when they are not needed at all. Therefore, it was expected that beep tones would have a positive effect on user performance and acceptance for interactions with complex utterances in general, and with complex utterances that contain an input example in particular, and a slight positive effect for the example condition for simple utterances.

3.7 Hypotheses

Based on the expectations described in the previous section our hypotheses were the following:

1. User performance will generally be higher for simple interactions.
2. User acceptance will generally be higher for simple interactions.
3. When the complex interactions are sufficiently simple, they will yield a higher level of user acceptance, comparable to or exceeding that of simple interactions.
4. Input examples will generally increase user performance, particularly for complex interactions.
5. User acceptance of input examples in simple interactions will differ strongly among people.

6. Beep tones as indicators of turn-taking will generally increase user performance for complex interactions, particularly complex-example interactions.
7. Beep tones will generally increase user acceptance for complex interactions, particularly for complex-example interactions.

3.8 Summary

Current IVR systems with a touch-tone interface aim for cost-effectiveness with a system-driven approach, while the ASR community is heading for complete user freedom through the use of recognition of conversational speech. An evaluation of the balance between these two extremes is necessary to determine optimal efficiency and user acceptance, now that speech is increasingly becoming a solution to the limitations inherent to touch-tone. While both user acceptance and interface efficiency may benefit from a more complex interaction, they will suffer from this complexity when speech recognizers cannot optimally deal with it or users do not feel comfortable with it. Speech recognition accuracy and user comfort can be improved by implicit curtailment of the user input. Different aspects of prompting that have an influence on this curtailment, such as utterance complexity, input examples, and cues for turn-taking, were evaluated in this study. It was expected that complex interactions with input examples and beep tones would lead to the optimal combination of user acceptance and user performance.

4 Method

An elaboration on the design and implementation of the study: the subjects, the procedure, the experimental design, and the materials and apparatus used

The design of the experiment application was directly derived from the design of the experiment itself. Therefore, its setup and functionality differed from what would be ideal for a real-life system. Efforts were made, however, to let the system appear as a ready-to-deploy telephone information system.

This chapter is organized as a comprehensive Methods section of a typical research article. In the first two sections, the subjects of the experiment and its procedure are described. The succeeding section outlines the experimental design. The concluding section, before the summary, contains an elaboration on the materials and apparatus used for the experiment.

4.1 Subjects

Thirty-two subjects were recruited by posters[10], put up in public places such as subway stations and shopping malls. None of the subjects worked for speech recognition, telecommunications, or market research companies. All were at least 18 years of age and had used an automated telephone information system in the twelve months prior to the experiment. Of the eighteen male subjects, eleven were students or younger than 30, four were between 30 and 40 years old, and three were over 40. Of the fourteen female subjects, eight were students or younger than 30, three were between 30 and 40 years old, and three were over 40. Subjects were paid twenty dollars for their participation in the experiment. The experiment was conducted on PureSpeech premises and took about forty-five to sixty minutes.

4.2 Procedure

The subjects came to PureSpeech premises for testing. They were told that they would help test some new telephone information systems, and that the data would be used to improve on these systems.

The subjects were seated in an enclosed room at a desk with a telephone. In front of them they had a booklet containing eight short scenarios. Each subject was given the same basic instructions on how to call in to the test system for each scenario, and what to do in case of misrecognition, or system errors—see *Appendix F: Subject Instruction* for the complete text. It was explained to them that misrecognition and system errors were not part of the experiment. They were also told that the experiment was not a test of how they themselves performed, but that it was testing the performance of the system.

They were then asked to read the first scenario carefully—once or twice, until they could imagine themselves in the described situation—and then call in to the system. The system would ask them for an access code, given to them on a separate answer sheet.

---

[10] The poster is included in *Appendix B: Recruitment Poster* on page 62.

1. Subject instruction
2. Go to first scenario
3. Read scenario
4. Call in to system
5. Interact with system
6. Write down answer
7. Check off rating scales
8. Go to next scenario
9. Repeat from step 3 till last scenario
10. Fill out global evaluation form
11. Post-Session Interview

Table 4.1—Experiment procedure

Punching in the access code led the subject to the right system and condition. The subjects would interact with the system by talking to it via a telephone handset. Once they had the information they were looking for, they would hang up, and fill in the answer on the answer sheet. After each of the eight scenarios in the booklet was a page with thirteen rating scales. Subjects were asked to put a check mark anywhere along the line of each scale, according to how they felt about the last system they called into. Each scale had two bipolar descriptive words on either side. After the last scenario and rating scale, the subjects were asked to place the four systems on a line to indicate relatively how much they liked each system.

The interactions of the users with the system were recorded on audio tape, and logged by the computer. After the subjects had finished the eight scenarios and the final evaluation form, a short interview was conducted, in which a set of four questions was asked. The first two questions dealt with their preference for utterance complexity and input examples. The third and fourth question asked whether it was always clear to them what they could say to the system, and what misrecognitions were encountered during the sessions.

4.3 Experimental Design

The experiment used a $2 \times 2 \times 2$ mixed design[11]—see Table 4.2. Utterance complexity—simple versus complex—and the presence or absence of an example were within-subject factors. The presence or absence of a beep tone was the between-subject factor. For the assessment of effects analyses of variance were conducted with a 5% significance-threshold p value for two-tailed tests and 10% for one-tailed tests.

4.3.1 Independent Measures

4.3.1.1 Utterance Complexity

For each scenario, subjects had to provide the experiment system with four pieces of data. In the simple utterance condition subjects were prompted for these pieces of data in four successive steps. At each step the system was able to recognize one piece of data. In the complex utterance condition they were prompted for the input in only two

	no beep tone		beep tone	
	no example	example	no example	example
simple utterance				
complex utterance				

Table 4.2—The 2 × 2 × 2 mixed experimental design

[11] A $2 \times 2 \times 2$ mixed design contains three binary variables, of which one or two are within-subject variables and the other one or two are between-subject variables.

successive steps. In these steps the system needed two pieces of data from the user in a single phrase of speech input.

4.3.1.2 Examples

In the example condition every request for input by the system ended with an example of how to phrase that input. This example served as a template that indicated the expected structure of the input. In the commuter rail system, for instance, the prompt asking for station names was: "From which station *to* which station would you like to travel? For example: from Prides Crossing to Salem." In the non-example condition the examples were simply omitted from the prompts[12].

4.3.1.3 Beep Tones

The system was not able to detect input while playing a prompt because a 'barge-through' feature was unavailable for the recognizer used. Therefore, subjects had to wait till the end of the prompt before speaking. In the beep tones condition, a short beep tone was played at the end of every prompt—*i.e.* at the end of the request for input in the non-example condition and at the end of the example in the example condition—as an indicator of turn-taking. Beeps were not played in the no-beep tone condition.

4.3.2 Dependent Measures

4.3.2.1 User Performance

With respect to user performance the research focused on task completion time, error rate and task compliance. Prompt playing time was included in total task completion time, since it was an integral part of the treatment—see also section 5.1.3 of the discussion of results. Two types of errors could occur: system errors and user errors, of which the latter one was of interest to this experiment. User errors were a dependent variable in the sense that user errors translate into longer task completion time and into an increased number of recognitions by the system for completion of the task. Task compliance was a control variable. This variable was used to exclude subjects that did not complete the tasks as required.

4.3.2.2 User Acceptance

Following each scenario, subjective user acceptance of the system just used was assessed with a set of thirteen bipolar adjective rating scales. These scales consisted of an overall 'Unacceptable—Acceptable' scale and the twelve individual adjective scales shown in Table 4.3. The choice of adjective pairs was based on a scale developed by Coleman (1985; as modified by Casali *et al.*, 1990). The rating scales were presented to the subjects with positive and negative adjectives placed on the right and left side of horizontal lines respectively[13].

Slow	——	Fast
Inaccurate	——	Accurate
Inconsistent	——	Consistent
Irritating	——	Pleasing
Undependable	——	Dependable
Unnatural	——	Natural
Incomplete	——	Complete
Uncomfortable	——	Comfortable
Unfriendly	——	Friendly
Distracting	——	Facilitating
Complicated	——	Simple
Useless	——	Useful

Table 4.3—Individual rating scales for assessment of user acceptance measure

4.3.3 Counterbalancing

In ASR studies of Capindale *et al.* (1990) and Biermann, Fineman & Heidlage (1992), learning effects were found over successive sessions. To make sure that neither learning

[12] Please see *Appendix C: Prompts Overview* for an overview of all prompts of the four systems.

[13] Please see *Appendix H: Bipolar Rating Scales* for the rating scale sheet.

effects nor the type of system would introduce biases in our experiment, and to ensure that every system was combined with every experimental condition, counterbalancing was necessary.

Counterbalancing was done according to a Latin Square[14]. The order in which the four systems were presented was counterbalanced over all subjects. Also counterbalanced was the pairing of the four different types of systems with the within-subject conditions. This produced sixteen different combinations in each of the beep and no-beep conditions. This way, the combination of four systems, times two utterance-complexity conditions, times two example conditions, times two beep tone conditions, added up to thirty-two different experimental setups.

Each of the thirty-two subjects was randomly assigned to one of these setups: all thirty-two subjects interacted with all four systems, with a different experimental condition for each of the systems. For each system, the subjects worked on two different tasks in the context of one of the four conditions: simple utterance, simple utterance with an example, complex utterance, and complex utterance with an example

4.4 Materials and Apparatus

A personal computer ran a single speech recognition application that was built especially for this experiment. This application was written in a proprietary scripting language, and contained all four systems. Subjects called in to this setup over a standard analog telephone line and spoke to the computer through an ordinary telephone handset. The computer was hooked up to the telephone line via a voice modem. A tape-recorder, connected along the line to the computer, started and stopped recording each session with the start and end of each phone call.

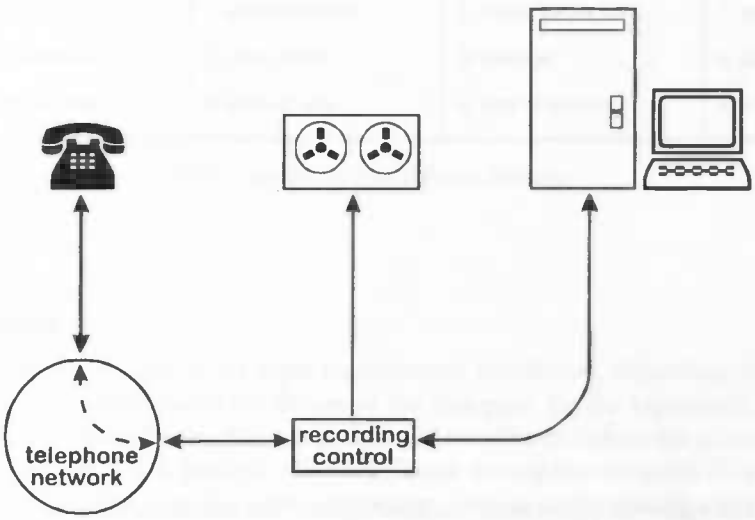


Figure 4.1—Hardware setup for experiment

[14] Please see *Appendix E: Subjects and Access Codes Overview* for an overview of the counterbalancing for this experiment.

Each subject went through all four beep or non-beep conditions. To decrease the learning effect of interacting with the system over succeeding tasks, subjects communicated with a different system for each of the four conditions. The difference between these systems was the domain for which they provided information. Thus, four mutually alternative systems were built into the application.

While their functionality was the same, their respective domains were different: commuter rail schedule information, auto loan calculation, auto blue book[15] value look-up, and catalog ordering. These domains seemed suitable for the desired setup. Table 4.4 provides an overview of the pieces of information required for each of the systems.

The commuter rail system provided a train number with the corresponding departure and arrival times, the auto loan calculator specified the monthly payment in dollars, the auto blue book gave the value of the specified car, and the catalog order system specified the total price of the order. A few randomly chosen examples of possible input are "Twelve thirty PM" for the departure time for the commuter rail system, "Five years" as the loan period for the second system, "Nineteen ninety six" or "Ninety six" as the year of the car for the third system, and "Sweater" as an item name for the catalog order system.

In the complex conditions, two such pieces of information were prompted for at the same time. To make sure that any possible domain bias would not influence the results, conditions were counterbalanced over domains, and so was the order of domains.

COMMUTER RAIL SCHEDULE INFORMATION	AUTO LOAN CALCULATION	AUTO BLUE BOOK VALUE LOOK-UP	CATALOG ORDERING
1. departure station	1. price of the car	1. make of the car	1. item name
2. arrival station	2. down payment	2. model of the car	2. quantity
3. departure time	3. loan period	3. mileage	3. size
4. day of the week	4. interest rate	4. year of the car	4. color

Table 4.4—Overview of input data for the four different domains

4.4.2 Call-Flows

The system led to one of the eight experimental conditions, depending on the last digit of the access code[16]. Since the design of the dialogues for the experiment application were to be deducted from these experimental conditions before the actual coding, it was beneficial to do a task analysis using a dialogue description notation (Dix *et al.*, 1993). Diagrammatic dialogue description notations, such as *state transition networks* (STN)—in which circles indicate states and arrows indicate transitions, are much used in dialogue design (Dix *et al.*, 1993). Although STNs allow for an overview of the structure of the dialogue, they lack significant descriptive power. Therefore, dialogue design was conducted by the use of call-flows, a similar notation that incorporated the descriptive

[15] The term *blue book* is generally used to refer to standard tables of depreciation.

[16] The mapping of the access code to the experimental conditions and the subjects is described in *Appendix A: Implementational Choices.*

power of ordinary flow-charts—see also section 2.2.7. With a clear overview of the number of interaction steps, and description of the data users provide at each state, these call-flows corresponded to a classic task analysis. The dialogues for the different experimental conditions were thus defined, before they were implemented by coding.

For the experimental conditions, two basic types of call-flow structure were used: simple and complex. From these two structures all other conditions were derived, either by adding examples or beep tones to the end of the prompts, or both. The call-flow for the simple utterance condition consisted of four steps (ASM1 to ASM4), as shown in Figure 4.2. At each step the caller was prompted for a certain piece of information. If the

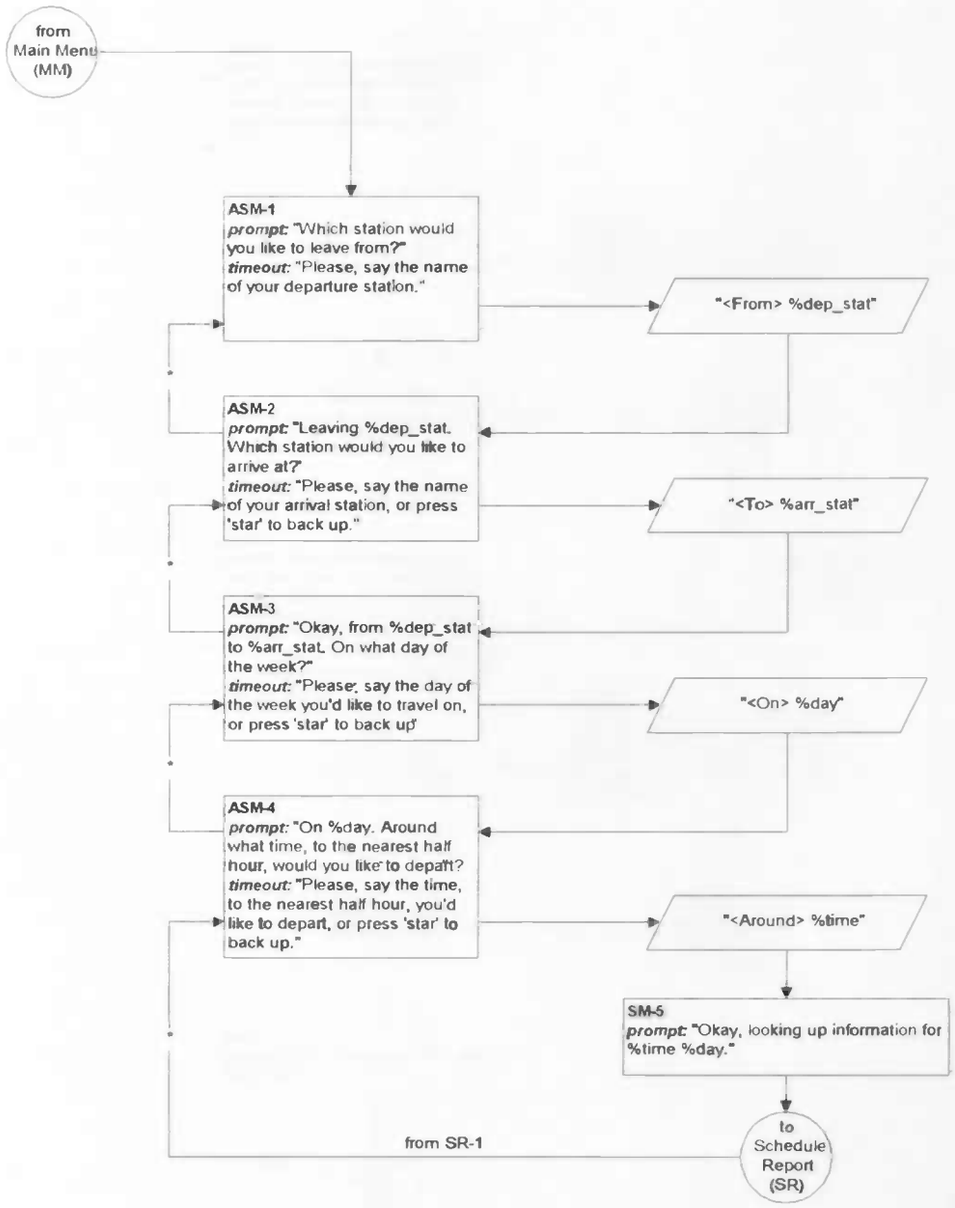


Figure 4.2—Call-flow for the simple utterance condition

input was recognized, the system went on to the next step and asked for the next piece of information. (For simplicity, this check on the input is not included in the call-flow in Figure 4.2) The call-flow for the complex utterance condition consisted of two steps (CSM1 and CSM3), as shown in Figure 4.3. At the first step, the user was prompted for the first two pieces of information. If this input was recognized, the remaining two pieces of information were prompted for in the second step.

When the system recognized all the input data, it used this to search or calculate the appropriate information. This information was then played to the caller. After playback, several options were presented to act on the information provided. Further interaction following from these options would not be incorporated in the data analysis;

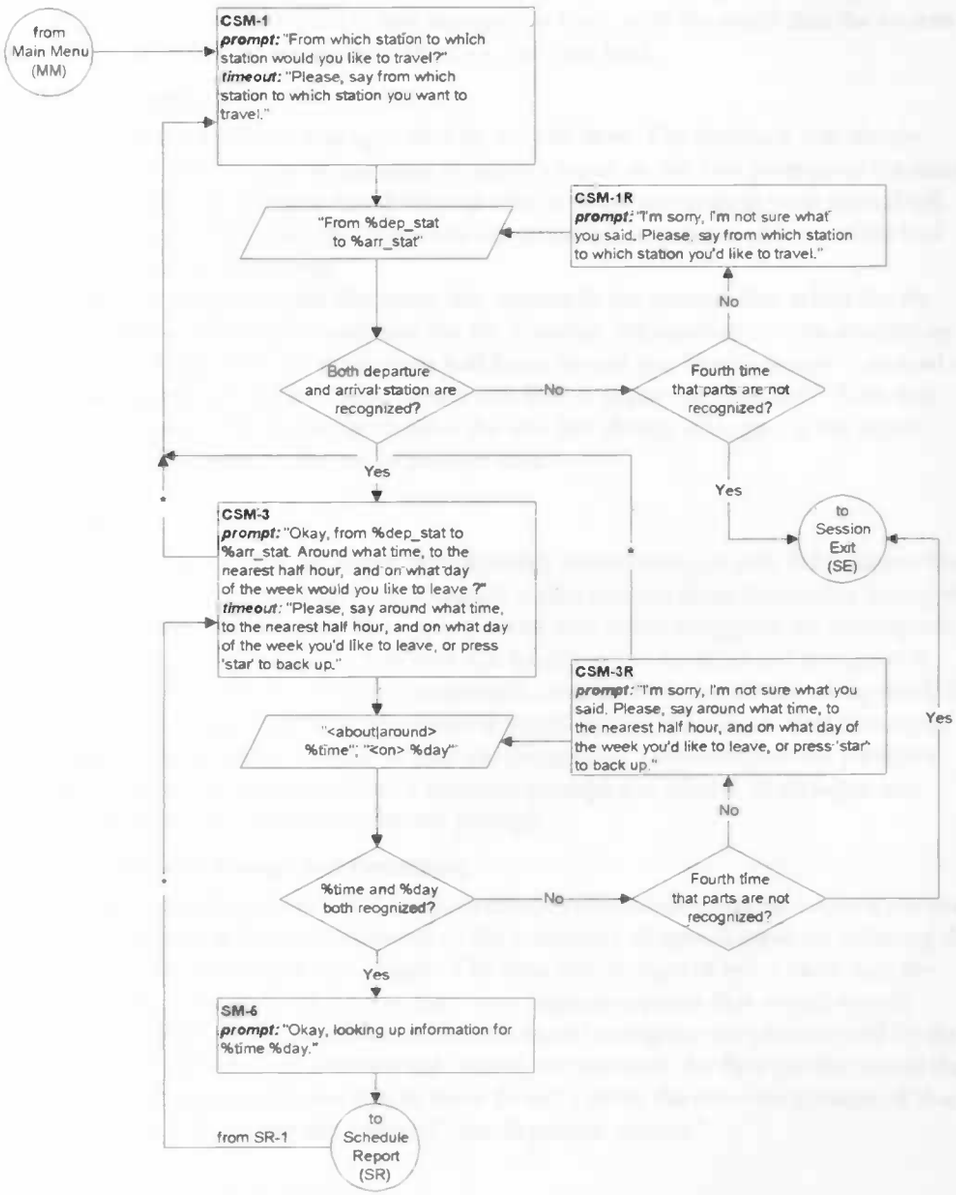


Figure 4.3—Call flow for the complex utterance condition

their mere purpose was to provide the system a more realistic appearance. One of the options was to repeat the information. The other options differed somewhat per system. After information playback in the loan calculator, for instance, subjects could request a new calculation for a different period or down payment, while the options in the auto blue book system, for example, allowed for the specification of another year or different mileage. A call-flow specification of this menu for the commuter rail system can be found in Figure 2.4, on page 14.

4.4.2.1 Error Recovery

The star key of the telephone key-pad was included as a means to back up to the previous recognition state—indicated in the call-flows by arrows marked with a '*' pointing in the opposite direction. Thus the caller would proceed through the interface by normal conversational interaction with the system, and back up by pressing the star key. The star key could be used in any recognition state, with the result that the system returned to the previous recognition state, *i.e.* one step back.

4.4.2.2 Feedback

The timing of the feedback was specified by the call-flow. The feedback was always provided immediately upon recognition of subject input, in the first prompt of the next recognition state. All prompts that followed after another recognition state started off with the feedback. This way, the caller was informed as soon as possible on what had been recognized by the system.

The following example illustrates this approach: the prompt that asked for the departure time in the simple condition for the schedule information system resulted in: "On Monday. What time, to the nearest half hour, would you like to depart?", instead of: "What time to the nearest half hour would you like to depart on Monday?" With this setup, the user was able to decide to press the star key shortly after giving the input without having to listen to the entire prompt first.

4.4.3 Prompts[17]

Prompts make up the speech output that provides information, or asks information from the user. Every state in our experiment started with a prompt from the system asking the subject for certain information. The system could then either recognize the subsequent input, not recognize it because it was said too loudly or too soon, or not recognize it because it did not correspond to the recognition context. If the input was recognized, the system went on to the next state, otherwise it would inform the subject that that input was either too loud, said too soon, or just not recognized, depending on the particular case. When misrecognition occurred a rejection prompt was played. If no input was heard at all, the system played a time-out prompt.

4.4.3.1 Prompt Design and Recording

The first prompt of every state was *implicitly directive*; this meant using an implicit template that would covertly achieve curtailment of the variability of speech input by inducing the subject to model their input accordingly. The time-out prompt of every state, and the error messages, were *explicitly directive*; they were explicit requests that would overtly achieve curtailment of the variability of speech input by shaping the phrases said by the user. In state ASM1 of the commuter rail system, for instance, the first prompt asked the user: "Which station would you like to leave from?", while the time-out prompt of that same state said: "Please say the name of your departure station."

---

[17] Please see section A.4 of *Appendix A* for a further elaboration on the implementational decisions concerned with the prompts of the experiment application.

The prompts were polite, had a friendly tone, they were unhurried and of low-pressure. They were kept to the point and as brief as possible. A complete overview of the prompts for all four systems can be found in *Appendix C: Prompts Overview* on pages 63 to 66.

4.4.3.2 Feedback in Prompts

To ensure that the interface appeared as natural as possible, implicit confirmation was implemented in the experiment application. This was done by letting the system confirm the user input by repeating it immediately after the subject had given the input and prior to prompting for the next input. Unless the subject would respond to the feedback by pressing the star key, the system would assume it had recognized the input correctly and continue the interaction.

4.4.3.3 Examples in Prompts

All prompts in the example condition ended by giving the subject an example of how to say the input. This meant that for every recognition state, there were several examples available of the possible information input required, namely in the first prompt, the time-out prompt and the error prompts.

Overlap in the precise values for these pieces of information in the examples was prevented, however. Examples were different from each other, and also different from the data used in the scenarios.

Since every subject would go through one different experimental condition for each of the four systems, examples were kept fully consistent over the complex versus the simple condition. Consistency was maintained over the data used for the examples, and the type of prompt they appeared in.

4.4.3.4 Overview of Prompt Structure

Most of the prompts started with feedback on what was recognized in the previous state. No feedback was provided in the prompts of the first recognition state of the interaction, or the prompts of the state in which the system reported the requested information. Prompts between the example and non-example condition respectively differed in the presence or absence of an example at the end of each prompt. The most complex prompt structure could therefore be found in the prompts of the example condition of the recognition states that required one or two pieces of information from the caller. The most complex prompt structure consisted of respectively: (1) feedback, (2) a request for input, and (3) an example of possible input. For state CSM3 of the auto loan calculator, this prompt-structure led to the following prompt:

> "Okay, *%amount* dollars with *%down* dollars down. Over how many years at what interest rate? For example: over 5 years, with 6.7% interest."

If the user had specified a 19,000 dollar loan with a down payment of 2,000 dollars prior to this prompt, for instance, then the variable *%amount* would be "19,000" and *%down* would be "2,000".

4.4.4 Grammar

All input that could be recognized by the system was specified in recognition contexts. Every state of the interaction—corresponding to the rectangular boxes in the call-flows—had its own recognition context. Only what was defined in these grammar specifications could be recognized. All detected input that could not be matched to what was specified for the state in question would lead to a rejection, and a rejection-prompt would then be played to the user, asking them to repeat their input.

4.4.4.1 Grammar Size and Recognition Accuracy

Recognition accuracy had to be maximized, and therefore recognition contexts needed to be as small as possible. They were limited to the data provided in the scenarios that the subjects had to go through and to the data that was used in the example prompts of the system. Any input other than what was given in the scenarios and examples could therefore not be recognized.

A typical grammar for a recognition state would consist of five different options for input, since it had to allow for the data provided in each of the two scenarios, and for the examples given in the opening, time-out and rejection prompts. While a grammar for departure times in a schedule information system, for instance, normally would allow for input of the entire range of times, the grammar for departure times for the commuter rail system of this study was reduced to times with an hour specification from a limited set of hours, concatenated with an optional "thirty" for the minutes and "AM", "PM" or "o'clock".

Subjects would not be aware of this limitation, since all necessary data could be recognized. Incorrect input would either be correctly recognized if it was identical to one of the examples, misrecognized if it would have a certain similarity to one of the examples or data in the scenarios, or rejected otherwise. Extraneous speech not specified in the grammars, such as "I'd like to..." or "please", for instance, was ignored as long as the mechanism of phrase-spotting was able to detect the essential parts of the input.

4.4.5 Eight Different Scenarios[18]

Subjects used each of the four systems according to scenarios they were given prior to testing. They read a different scenario for every task. Two scenarios were written for each of the systems the subjects had to call into. This meant that each subject called in to the application eight times, once for each of the scenarios, twice for every system.

Two scenarios for one domain applied to the same system, but differed in the pieces of information that users were required to give as input. These two scenarios also differed in the situation described and in the order in which these pieces of information were presented in the text.

All scenarios described a situation in which the subject needed some information. The system would provide the information based on four pieces of data the subjects provided themselves. These four pieces of information were underlined and italicized in the text to make them stand out.

Subjects were encouraged to picture themselves in the described situation by the phrase "Imagine that you..." with which each scenario started. Instructions given to the subjects prior to the experiment[19] included reading the scenarios once or twice, until the described situations were clear in their minds. This was also repeated on the scenario sheets themselves, at the top of the page. Every scenario ended with a phrase of the form "Call the [...] System to find out..."

The first scenario for the commuter rail schedule information system, for instance, described a situation in which the subject would call in to the system to find out how to get to work in time on the first day of a new job. In the second scenario of the auto blue book system, the system is asked for information on the value of a used car a friend is interested in buying, to find out whether its owner is asking a fair price for it. All other scenarios described situations of similar simplicity.

The scenarios were of roughly the same length of under a hundred words. They were printed on separate sheets, and presented to the subjects separately. They were

---

[18] Please see also *Appendix D: Scenario Overview* on page 67.

[19] Please see also *Appendix F: Subject Instruction* on page 70.

bound in a booklet together with a working sheet for filling out the information provided by the system, sheets with evaluation scales for rating the systems after each session and an overall preference sheet for an general comparison between the systems at the end of the experiment[20].

4.5 Summary

The effects of utterance-complexity and the use of beep tones and examples in system prompts were studied in a mixed experimental design. Thirty-two subjects were recruited to help test telephone information systems using speech recognition. They called in to an experiment application that consisted of a commuter rail schedule information system, an auto loan calculator, an auto blue book value lookup system, and a catalog order system. They interacted with the speech application according eight different scenarios—two for each of the four systems—to obtain certain information. The interactions were logged by the computer and recorded on audio cassette. At the end of each of the eight runs, subjects rated their acceptance of the system just used with bipolar adjective rating scales. At the end of their session a short personal interview was conducted to obtain additional information on preferences and recognition problems.

[20] Please see *Appendix G: Experiment Work Sheet*, *Appendix H: Bipolar Rating Scales*, and *Appendix I: Experiment System Preference Form* on pages 71, 72, and 73 for the work, ratings scales, and overall preference sheet respectively.

5 Results

Analysis of user performance data and user acceptance data, and discussion of results

Data analysis was focused on user performance and user acceptance. In this chapter *user performance* data will be looked at first. This will be done by analyzing the number of recognitions that were necessary for the subjects to complete the scenarios, and by analyzing the total task completion times. *User acceptance* will be evaluated by analyzing the results of the bipolar rating scales.

The cell means of the number of necessary recognitions, total task completion time and the scale ratings will be shown in tables. The convention for the tables with cell means of effects for two-tailed tests in this chapter is that significant results are shown in a bold typeface while effects that are marginal are italicized. When marginally significant effects are consistent with the predictions, their significance will be evaluated using a one-tailed test.

5.1 Performance Data

5.1.1 Performance Data Analysis Procedure

Data on user performance was collected by automatic logging by the experiment application and by audio recordings of the sessions. The log files specified all user actions, such as entering of subject code or backing up, and all system actions, such as recognition or rejection, with time and date stamps included. The audio recordings contained both user input and system output[21].

For data analysis, system output and recognition outcomes were filtered out of the log files. Recognitions, rejections and backups were counted for each run. Total run time and total prompt playing time were also calculated for each run[22]. An analysis of variance (ANOVA) was made for the number of recognitions and for the task completion time. Results whose significance was affected by the inclusion of system-condition pairing are not presented here.

The number of recognitions needed per run was taken as a measure of user performance since the interactions with the experiment application required a certain minimum number of recognitions. Recognition amounts higher than the minimum indicated the occurrence of rejections or backups and therefore non-optimal user performance.

---

[21] The main purpose of the audio recording was to serve as a possible control facility for the quantification of the experimental data in case something in the log files would be unclear. This, however, did not occur.

[22] A *run* was a single session of one subject for one scenario, and was defined as the beginning of the prompt for the first recognition state to the beginning of the prompt that provided the requested information. When the subject backed up from the state where the information was reported, the run continued until the beginning of the next report, in which case the backup was counted and run time simply extended.

Total task completion time was the second measure for user performance. Five task completion times (out of 256[23]) deviated more than 3s from the mean and were replaced by the mean + 3s value.

5.1.2 Analysis of Number of Recognitions

The minimal number of recognitions that was required to complete a task differed for the simple-utterance and the complex-utterance condition. When no misrecognitions occurred and no backups were necessary, the interaction for the simple-utterance condition consisted of four recognition steps, and for the complex-utterance condition of two recognition steps. After a rejection or backup the system prompted the user again for the piece of data that it tried but failed to recognize. Therefore, any greater number of recognitions than the minimum indicated that user performance was not optimal.

CONDITION	RECOGNITIONS	TIME-OUTS	REJECTIONS	BACKUPS
simple	4.44	0.09	0.47	0.31
simple-example	4.59	0.18	0.25	0.53
complex	3.34	0.03	0.50	1.37
complex-example	2.38	0.16	0.44	0.38
simple-beep	4.84	0.06	0.31	0.69
simple-example-beep	4.66	0.06	0.34	0.50
complex-beep	2.63	0.06	0.25	0.75
complex-example-beep	2.28	0.06	0.22	0.25

Table 5.1—Mean number of recognitions, time-outs, rejections and backups for the eight experimental conditions

An ANOVA on the full model yielded a significant effect of 'beep tone' × 'utterance complexity', $F(1,24) = 4.73, p < .041$, and marginally significant effects of 'example', $F(1,24) = 3.39, p = .078$, and 'utterance complexity' × 'example', $F(1,24) = 3.32, p = .081$. These effects confirmed hypotheses 6, on user performance with beep tones, and 4, on user performance with input-examples, respectively, as described below (for all hypotheses, see section 3.7). No effects were found to support hypothesis 1, on higher user performance with simple interactions.

The 'beep tone' × 'utterance complexity' effect was slightly negative for simple utterances and positive for complex utterances. As Table 5.2 demonstrates, beep tones led to a 5% increase in the mean number of necessary recognitions for the simple-utterance condition and to a 14% decrease in the mean number of necessary recognitions for the complex-utterance condition.

	no beep tone	beep tone
simple utterance	**4.52**	**4.75**
complex utterance	**2.86**	**2.46**

Table 5.2—Number of recognitions: 'beep tone' × 'utterance complexity' interaction ($p < .041$)

It appears from these means that when the grammar was simple, an indication of turn-taking had

[23] Each of the 32 subjects went through eight scenarios, leading to a total of 256 runs.

no example	3.81
example	3.48

Table 5.3—Number of recognitions: overall 'example' effect (p = .078)

	no example	example
simple utterance	4.64	4.63
complex utterance	2.99	2.33

Table 5.4—Number of recognitions: 'example' × 'utterance complexity' interaction (p = .081)

a slightly negative effect. The prompts used in this condition were short enough to have a predictable ending, making it easy for the subjects to realize when they could start speaking. The slightly negative effect may have resulted from confusion when a beep tone was played at the moment subjects already initiated their response.

When utterance complexity was high, however, the beep tone reduced the number of recognitions required to complete the task. Two possible explanations could account for this effect. First, with longer phrases it might have been less obvious to the subjects when they could start speaking, and the beep tone clearly indicated the end of the prompt. There was a significant reduction in the mean number of rejections when a beep was added to the prompts in the complex conditions—see Table 5.1—indicating perhaps that subjects did not hesitate as much when starting to speak. Second, with longer prompts subjects might have had a stronger tendency to interrupt the prompt. The inclusion of a beep tone at the end of a prompt notified the subjects that they had to wait until the end of the prompt before speaking. Lower means for the number of rejections and backups for the complex-beep condition—see Table 5.1—supported the likelihood of this second explanation because speaking too early could lead to misrecognition or rejection since part of the input would get lost.

The 'example' effect, though only marginally significant, suggests that examples were helpful in general. Overall, the inclusion of an example in the system prompts may have led to the decrease in the number of necessary recognitions by approximately 9%—see Table 5.3. The advantage of using an example in a prompt may be limited to complex recognitions only, however. As shown in Table 5.4, the 'utterance complexity' × 'example' effect—although only marginally significant—would indicate that the simple-utterance condition did not benefit from the presence of examples, but that examples increased user performance by 22% in the complex-utterance condition. Since this was an expected result, this effect is significant (p = .041) when using a one-tailed test.

5.1.3 Analysis of Total Task Completion Time

If subjects had any difficulty with the task it was likely to cause them to take longer to complete it. They needed more time to think about what to say, and also to back up and rephrase their input. For subjects who had difficulty using the system, time-out prompts were also likely contributors to increased total task completion time. When subjects did not respond to the prompts within a certain time, the system would play a time-out prompt.

The total task completion time included prompt playing time. While it was also true that the example treatment increased task completion times by playing additional prompts—namely, the example prompts—this was an integral part of the treatment and therefore necessarily reflected in the time measurement.

Two significant interactions were found: 'beep tone' × 'utterance complexity', $F(1,24) = 8.62, p < .008$, and 'example' × 'utterance complexity', $F(1,24) = 5.33$, $p < .031$. These effects also confirmed hypotheses 6 and 4 respectively, as described below. Again, no effects were found to support hypothesis 1.

CONDITION	PROMPT TIME	INPUT TIME	TOTAL TIME
simple	53	34	87
simple-example	64	40	104
complex	64	59	123
complex-example	66	38	104
simple-beep	56	41	97
simple-example-beep	63	43	106
complex-beep	54	36	90
complex-example-beep	57	35	92

Table 5.5—Mean prompt playing time, user input time, and total interaction time for the eight experimental conditions

	no beep tone	beep tone
simple utterance	**96**	**102**
complex utterance	**107**	**98**

Table 5.6—Total task completion time: 'beep tone' × 'utterance complexity' interaction ($p < .008$)

	no example	example
simple utterance	**92**	**105**
complex utterance	**114**	**91**

Table 5.7—Total task completion time: 'example' × 'utterance complexity' interaction ($p < .031$)

In consistence with the results of the analysis of the number of recognitions, it appears that the beep tone had a minor negative effect on simple recognitions but produced shorter task completion times for complex recognitions—see Table 5.6 (cell means are expressed in seconds). The presence of beep tones increased total task completion time in the simple-utterance condition by 6%, while it made interactions in the complex-utterance condition 8.5% more efficient. This consistency with the results of the analysis of the number of recognitions is not surprising, since a larger number of recognitions was bound to lead to longer task completion times. Beep tones may also have decreased task completion times by eliminating the hesitation users may have felt when giving input.

In consistence with the predictions for this study as outlined in section 3.6, *Expected Results*, the presence of an example significantly helped complex recognitions, but did nothing for simple ones—see Table 5.7. With a mean difference of thirteen seconds, total task completion times were 14% longer for the simple-utterance condition when examples were included in the prompts. This effect was only partly due to the length of the example prompts themselves, since these added only eight seconds to the run in the simple-utterance condition for each of the four systems. The remaining five seconds of mean difference would have been the result of lower user performance in the simple-utterance condition with examples. Extra prompt-playing time needed for the examples in a single run of the *complex*-utterance condition varied from five to ten seconds for the different systems, with an average of seven seconds. Despite these seven extra seconds, examples caused mean total task completion times to be 23 seconds shorter for the tasks with complex recognitions. These results indicate that subjects found it more difficult to phrase complex utterances in the required format and

that they used the examples as a template for their input. The formulation of the simple input was too easy to benefit from the explicit templates provided by the examples.

5.2 Acceptance Data

5.2.1 Acceptance Data Analysis Procedure

Data on user acceptance was collected in two ways: with bipolar rating scales[24] marked off by every subject after each run, and with personal interviews conducted at the end of each session. The rating scales had keywords with a negative meaning on their left side and keywords with a positive meaning on the right. In order to quantify subject evaluation the marks were assigned a value from 1 to 16 according to their distance from the left side of the scale. Thus a larger scale value indicated a more positive attitude. These values were averaged per subject over the two runs for each of the four experimental conditions.

An ANOVA was conducted for the mean over all thirteen scales and for each of the scales separately. The analysis of user acceptance was averaged over scale items and scenarios. Subjects 7, 25, and 28 were omitted because their responses were almost entirely extreme values (1 and 16). The system-condition pairing was not included in the ANOVA model, although in cases where significant effects were found, it was evaluated whether the effect interacted with the system-condition pairing.

5.2.2 Analysis of Rating Scales

No interaction effects were found between the different independent variables for the rating scales. No significant effects were found for the mean of all rating scales. For the thirteen rating scales separately, no significant results were obtained for the eight of the thirteen rating scales: 'Unacceptable—Acceptable', 'Slow—Fast', 'Inconsistent—Consistent', 'Irritating—Pleasing', 'Unnatural—Natural', 'Incomplete—Complete', 'Complicated—Simple' and 'Useless—Useful'.

Marginal effects of 'utterance-complexity' were found for the rating scales 'Inaccurate—Accurate' and 'Unfriendly—Friendly'. Significant effects of 'utterance-complexity' were found for 'Undependable—Dependable' and 'Distracting—Facilitating'. These effects validated hypotheses 2 and 3, on user acceptance of simple versus complex interactions. No results were found to support either hypothesis 5, on the acceptance of input-examples, or hypothesis 6, on the acceptance of beep tones. An opposite example-effect, however, became apparent in the analysis of the 'Uncomfortable—Comfortable' rating scale. All effects, per rating scale, are described below.

simple utterance	*10.15*
complex utterance	*11.60*

Table 5.8—'Inaccurate—Accurate': 'utterance complexity' effect (p = .073)

Inaccurate—Accurate

There was only a marginal effect of 'utterance complexity' for the 'Inaccurate—Accurate' rating scale, $F(1,27) = 3.48$, $p = .073$. Complex systems were considered slightly more accurate than the simple ones—see Table 5.8. This was most likely due to the fact that there were more opportunities for the system to make an error in the simple-utterance condition, since these interactions consisted of twice as many steps as the complex-utterance condition.

[24] Please see *Appendix H: Bipolar Rating Scales* on page 72.

simple utterance	**9.49**
complex utterance	**11.41**

Table 5.9—'Undependable—Dependable': 'utterance complexity' effect ($p < .005$)

simple utterance	*11.70*
complex utterance	*12.50*

Table 5.10—'Unfriendly—Friendly': 'utterance complexity' effect ($p = .077$)

simple utterance	**9.50**
complex utterance	**10.50**

Table 5.11—'Distracting—Facilitating': 'utterance complexity' effect ($p < .044$)

no example	**10.49**
example	**9.97**

Table 5.12—'Uncomfortable—Comfortable': 'example' effect ($p < .009$)

Undependable—Dependable

A significant effect of 'utterance complexity' was found for the 'Undependable—Dependable' rating scale, $F(1,27) = 9.86, p < .005$. Complex systems were considered somewhat more dependable than simple systems—see Table 5.9. Again, this may have reflected the greater perceived accuracy of complex systems.

Unfriendly—Friendly

A marginally significant effect of 'utterance complexity' was found for the 'Unfriendly—Friendly' rating scale, $F(1,27) = 3.38, p = .077$. Complex systems were thought to be slightly more friendly than simple systems—see Table 5.10. This effect became marginally stronger , $F(1,21) = 3.90, p = .062$, with the inclusion of system-condition pairing in the model, but did not interact with system-condition pairing, $F(3,21) = .66, p = .588$.

Distracting—Facilitating

A significant effect of 'utterance complexity' was found for the 'Distracting—Facilitating' rating scale, $F(1,27) = 4.51, p < .044$. The subjects rated complex-utterance systems slightly more facilitating than simple-utterance systems—see Table 5.11. This effect lost some strength and became marginal, $F(1,21) = 3.95, p < .061$, when system-condition pairing was included in the model. No significant system-condition pairing × 'utterance complexity' interaction, $F(3,21) = .020, p < .997$, was found, however.

Uncomfortable—Comfortable

A significant effect of example was found for the 'Uncomfortable—Comfortable' rating scale, $F(1,27) = 8.25, p < .009$. Users were slightly more comfortable without examples than with examples—see Table 5.12. Section 5.2.3 gives an insight into the possible reasons for this finding.

5.2.3 Post-Session Interviews

In the post-session interviews, subjects freely commented on what they liked and disliked about the systems. They commented on certain aspects of the implementation, and on the usefulness of implementing the domains of the four systems into an automated telephone information system. A number of observations are worth mentioning.

First, most subjects appreciated the use of ASR in automated telephone systems. Although some people were already interested in speech recognition and saw this study as an opportunity to experience it first-hand, others who had not heard of it before were pleasantly surprised by the usefulness of ASR. When people were not pleased with the speech recognition, this was due to low recognition accuracy.

Second, it turned out that the subjects had very little tolerance for misrecognition. Several subjects explicitly complained about the need to back up and rephrase input.

Third, those subjects who disliked the entire experiment application in general, felt this way because they did not like automated telephone systems in general. One subject explained that, for the first scenario, he started to say something he was certain the system would not recognize, in the hope to be transferred to an operator.

Fourth, it became apparent that large individual differences existed with respect to different aspects of the system. The appreciation of examples in the system output, for instance, differed considerably among subjects. While some people stated that the examples were helpful or at least did not annoy them, others found the examples offensive because they felt they were being patronized. Large individual differences were also found in the appreciation of the overall ease of use of the systems. While some people stated that they had no problems whatsoever in going through the scenarios, others were sometimes not sure what to say. Overall, some people were indifferent to utterance complexity, examples, or beep tones, while others stated a clear preference for specific features under certain conditions—*e.g.*: examples only in complex conditions. Yet others had more general preferences—*e.g.*: examples for all conditions.

Fifth, most subjects were unaware of the experimental manipulation. They did not consciously realize the differences in utterance complexity, or that some systems contained examples or beep tones while other systems did not, until they were asked about it during the interview.

In spite of the large individual differences that existed between subjects with respect to their overall acceptance and acceptance of certain features of the different experimental systems and conditions, the interviews confirmed that high levels of recognition accuracy are necessary at all times for good user acceptance and performance.

5.3 Summary

The results for the total task completion times were consistent with the outcomes of the analysis of the number of recognitions. The effects of the 'example' treatment and 'beep tone' treatment were similar.

The addition of examples and beep tones to the system prompts led to increased user performance for the complex-utterance condition. Despite longer system prompts less time and fewer recognitions were necessary to complete the tasks. Interactions involving simple utterances did not benefit from either examples or beep tones, or became somewhat less efficient.

No significant effects were found for the mean over all rating scales or for the specific 'Unacceptable—Acceptable' scale. For four individual scales, however, significant or marginally significant ($p < .08$) 'utterance complexity' effects were found. With the 'Inaccurate—Accurate', 'Undependable—Dependable', 'Unfriendly—Friendly', as well as with the 'Distracting—Facilitating' rating scale, subjects rated a preference for complex systems over simple systems. A significant 'example' effect was found for 'Uncomfortable—Comfortable'. Systems without examples were preferred to systems with examples.

6 Discussion and Conclusion

Interpretation of study results: the implications of this study for the design of speech interfaces for IVR systems, and suggestions for future research

Based on the results described in the previous chapter, the implications of this study are evaluated here in the first section. The limitations of the experiment will be outlined in the second section, which will lay the ground for suggested improvements on the design of this study described in the third section. Proposals for further related research—in the fourth section—are based on both the implications and limitations of this study.

6.1 Study Implications

The findings of this study on user performance and user acceptance as a result of the 'utterance complexity', 'example', and 'beep tone' treatments can be summarized in the following six points:

- User performance in interactions with simple utterances was not affected by the inclusion of input examples or beep tones in the system output.
- User performance in interactions with complex utterances was improved by the inclusion of input examples in the system output.
- User performance in interactions with complex utterances was higher for systems that played beep tones as an indication of turn-taking.
- User acceptance was higher in systems that used a complex grammar for the interaction as opposed to systems that used a simple grammar.
- User acceptance was higher in systems that did not play input examples as opposed to systems that did.
- Large individual differences were apparent in user acceptance of input examples.

6.1.1 Naturalness of Interaction

The results indicate that most people—or at least the majority of participants in this study—appreciate the natural interaction that speech recognition interfaces provide. Subjects had an overall preference for systems that did not play examples of possible input. Since examples are not present in a natural human interaction, this may explain why subjects disliked them in the systems they tested. As elaborated in section 5.2.3, *Post-Session Interviews*, subjects explicitly stated their appreciation of ASR. The higher ratings subjects gave to systems that made use of complex utterances were also an indicator of their preference for a more natural interaction. This would follow from the fact that interaction using utterances that have a certain complexity is more closely related to the naturalness of human discourse than interaction by using single-word phrases.

The expression of subject preference for complex utterances in the accuracy and dependence ratings—and not so much in the naturalness rating—may also indicate,

however, that it is not the naturalness of speech but a greater perceived competence of the system that makes a smaller number of steps desirable.

On the other hand, the apparent user indifference with respect to the naturalness of the systems was an indication in itself of the naturalness of the systems. As outlined in section 5.2.3, most subjects were unaware of the experimental manipulation. Just like people are unaware of the complexity of natural inter-human discourse, the subjects did not realize that some systems used more complex utterances. They were satisfied when the system quickly provided them with the information they desired, without paying attention to what type of utterance was used to obtain the information. The capabilities of the system to recognize their complex speech were of no interest to them since their aim was merely to obtain the information the scenarios required, and report any obstacles or inconvenience encountered in the testing process. As it now turned out, where complex utterances might have introduced difficulties, they were not even noticed by the subjects. This indicates their naturalness and ease of use in speech interfaces of automated telephone systems.

6.1.2 Hybrid Interface Structuring

The higher user acceptance and apparent naturalness of complex-utterance systems would indicate that in designing a speech interface it is best to strive for a low number of steps in the interaction and for the use of a grammar that specifies phrases more complex than single words. As it appears from the results of this study, however, examples also need to be provided to serve as templates for the complex input. While examples were not helpful with simple utterances, they significantly improved user performance in complex interactions—see section 5.1.3. Beep tones were also found to improve user performance in the complex-utterance condition—see sections 5.1.2 and 5.1.3. When the interaction with an automated system reaches a certain complexity users need help with formulating their input. Also, cues about the interaction itself—such as beep tones for turn-taking—are necessary to improve robustness of the interface.

It is debatable, however, whether it is the right approach to extend prompts with examples to all complex-utterance speech interfaces. In this study, the post-session interviews clearly revealed large individual differences, particularly with respect to the appreciation of examples. To allow for this diversity it would be best to present examples only to users who are likely to need them. An ideal solution would be a mechanism that detects the best suitable way of prompting for the specific user and changes prompting techniques accordingly. Large individual differences make a great deal of flexibility necessary. Flexibility can be achieved by a hybrid interface structure that incorporates several degrees of utterance complexity and provides examples only when necessary.

6.1.3 Recognition Accuracy

A design as flexible as the one described in the previous paragraph could only lead to acceptable results when high levels of recognition can be maintained, however. User acceptance of low recognition accuracy is minimal, and misrecognitions make an interaction inefficient. While ensuring high recognition accuracy is an essential element in any speech application, it is Human Factors considerations that can combine high user acceptance and high user performance into a single speech application. This can be done through the use of complex utterances that convey multiple pieces of information, a small number of interaction steps, a flexible application of examples to provide guidance on possible input, and conversational cues for guidance of the interaction.

6.2 Study Limitations

There are five study limitations that need to be taken into account when drawing conclusions on the results of the data analysis of this study.

First, as described in paragraph 4.3.1.1, the speech application built for this experiment needed four pieces of data in order to provide certain information or a service. These four pieces were prompted for either separately in four steps, or in pairs in two steps. The question now is to what extent the findings of this study can be generalized to systems that have a different setup. Performance results will depend, for instance, on the way two or more necessary pieces of data combine into a single utterance. When a system requests for multiple pieces of information in a single utterance that do not combine very well intuitively, it may be more difficult for users to phrase the required input.

Second, in the data analysis of the study, some latitude was given to the task of the first Commuter Rail scenario because this scenario was more difficult than the others. The scenario specified an arrival time, while the application requested the user to specify a departure time. To find the right train, users could ask for information about earlier or later trains after the system had reported the train schedule for the specified input data. Glancing through different train schedules was not part of the study, however. Therefore, once the users had given initial input for the first Commuter Rail scenario that was within a three hour range of the correct train, extra total run time and further recognitions were not counted in the data analysis. Only successfully completed scenarios—possibly done by scanning different trains—were rated successful and incorporated in the data analysis. Had the range used been narrower than three hours, it would falsely have reflected into lower user performance results for the first Commuter Rail scenario which would have decreased the overall user performance rate. On the other hand, the three hour range allowed subjects to correct errors in a way that would not be incorporated in the data analysis—namely, by scanning through different trains. However, since subjects did not know this option when starting the first Commuter Rail scenario, they were more likely to correct their errors by backing up and rephrasing their input. Therefore, it was assumed that there would be no bias from the latitude given to the first Commuter Rail scenario.

Third, recognition accuracy and speed were unstable during the sessions due to technical difficulties with the line modem that was used. While recognition was fast and accurate on most occasions, it could be slow and inaccurate on others, independent of the user input. Instabilities were also introduced by the need to reboot the system on several occasions when the modem had caused the system to 'hang'. These instabilities were frustrating to the users and were a likely cause of larger variability and lower means of the acceptance and performance data.

Fourth, speech technology is continuously improving—the speech recognizer used for this experiment had clear limitations when compared to newer versions that have become available during the course of the study. While it was necessary at the time the study was conducted to keep recognition contexts to an absolute minimum and while users could not start speaking before the end of the system-prompts, owing to the weak recognizer used, PureSpeech has improved recognition accuracy levels considerably in the meantime and has made 'barge-through' a properly functioning facility. Improved recognition accuracy should have a positive effect on both user acceptance and performance. The possibility for users to interrupt system output may have significant effects on the course of the interaction—the use of beep tones will be made superfluous, for instance.

Fifth, the 'utterance complexity' effect and the 'utterance complexity' × 'example' interaction of the user performance data were not the same for all system-condition pairings. Although the system-condition pairings were counter-balanced, and the results generalized over all four systems, it was evident that the domains of speech applications had a direct influence on the way the speech interface should be structured.

6.3 Suggested Improvements on the Design of this Study

This study evaluated utterance complexity on two levels. The simple level consisted of utterances that contained a single piece of information of one or two words—*e.g.*: "From Salem". The complex level consisted of phrases containing two pieces of information of about two to four words—*e.g.*: "From Salem to Ipswich". Expansion of the experimental design with one or more levels of utterance complexity, however, may provide a richer set of data for this treatment, covering a wider range of complexity. Since the ASR community is constantly pushing recognition into the direction of highly complex speech, it could be beneficial to examine user responses to very open-ended prompts, such as "How may I help you?".

Other improvements on the design of this study are concerned with the developments of speech recognition technology. Currently available are speech recognizers that achieve better accuracy than was possible in this study, and have important functioning features like 'barge-through'. Both improved recognition accuracy and the 'barge-through' facility call for a change of the experimental design of this study.

When users are permitted to interrupt system output, beep tones are no longer necessary as an indication of turn-taking, since an exact moment when the user can start speaking no longer needs to be specified. While 'barge-through' urges for the beep tones to be eliminated from the speech interface design, other natural conversational cues, such as change in intonation, could perhaps be of interest as a more effective replacement.

Unstable and imperfect recognition was a likely cause of the impedingly large variability in the experiment data of this study. Some significant effects may have been obscured by this variability, which a study with a new and more powerful recognizer could have made more apparent. On the other hand, even a recent recognizer may be insufficiently powerful for a study examining the effects of higher levels of utterance complexity. A possible solution to this problem could be, for instance, to develop a 'Wizard-of-Oz'[25] type of setup for the complex recognitions. Another solution could be to use a speaker-dependent speech recognizer and let the subjects go through a voice training session prior to the experiment to bring recognition accuracy to optimal levels. Although the results of such a design may not be directly applicable, they would be invaluable when the strength of speech recognizers has further improved and speech interfaces are designed to allow a large amount of user freedom.

6.4 Suggestions for Future Research

6.4.1 Hierarchical Utterance Complexity

An in-depth study is needed to examine how simple and complex grammars can be combined in a single speech interface structure and in what ways such a combination would affect user performance and acceptance. Such a study should focus on the possibilities and usefulness of a hierarchical interface structure with several layers of utterance complexity.

[25] Please see footnote on page 26 for an explanation of the principles of 'Wizard-of-Oz'.

It would be useful to see whether the particular advantages of both complex and simple utterances could be exploited to make the interface optimal for a wide range of users under most circumstances. The advantages of complex phrases are the smaller number of interaction steps necessary and the possibility of combining multiple information into a single utterance. The advantage of simple utterances is that they are easier to formulate and they do not pose so much stress on human resources.

A hybrid system that incorporates both simple and complex utterances could perhaps start with prompting the user for complex input, which could be a single utterance containing all the necessary pieces of data. An example of such input for the Commuter Rail system would be: "I'd like to depart from North Station to Chelsea on Friday, at 5:30 PM." If the system would not recognize all the pieces of data in the complex response, it could then ask for the missing pieces with simpler utterances—*e.g.*: "On what day of the week?"—until it has all the data it needs.

6.4.2 User Sensitive Prompt Adjustment

Related to the need for research on the combination of simple and complex grammars is the need for a study on user sensitive prompt adjustment. The provision of examples, for instance, would ideally be tailored to the behavior of the user and the complexity of the interaction.

As shown in section 2.2.5, examples and other kinds of help are only useful when they are needed. Mechanisms that detect the best ways of prompting for a certain user, and make it possible to change prompting techniques accordingly, can prevent irritating users who do not need help, and provide ready access to assistance to users who do. Brems *et al.* (1995) found that careful selection of the pause duration between a prompt and the time-out prompt can ensure that the help provided in the time-out prompt is only heard by users who are likely to need it.

The choice of utterance complexity is another aspect of prompting that could be adjusted during the interaction.

6.4.3 Sociolinguistics

More extensive research on the applicability of sociolinguistic findings on human-to-human discourses in a speech recognition environment might be invaluable at a later stage, when recognizers allow users such freedom that the smoothness and efficiency of natural speech can be effectively exploited in ASR applications. User comfort and acceptance are especially likely to benefit from this approach. With respect to the current level of necessary limitation of recognition contexts, sociolinguistics may provide insights on ways of maintaining user comfort with curtailment of user input.

6.4.4 'Barge-Through'

The results of the current study could be further fine-tuned in an experiment using a more powerful recognizer. Comparing the data of the two studies could indicate a pattern for future developments in ASR and speech interface design. Stifelman, Arons, Schmandt & Hulteen (1993) already concluded that 'barge-through' would be an indispensable facility. The possibility of interrupting audio output doubtless has a large impact on speech interface design.

6.4.5 User Instruction and Error Correction

Further research on user instruction and error correction mechanisms for auditory interfaces is needed. In this study, the function of the star key was clearly mentioned in the instructions, and it was mentioned in the opening prompt of the experimental system as well. Nevertheless a number of subjects failed to remember this option, and were

unable to correct recognition errors. This implied that user instructions should be very clear, and that the way to correct errors resulting from missed or lacking instruction must be very intuitive. Baber & Hone (1993) concluded that it is inappropriate to apply just one type of error recovery mechanism. They discerned a relationship between recognition accuracy and error correction strategy. Ainsworth (1993) affirms that with respect to error recovery mechanisms a choice strategy leads to reduced transaction times for low recognition rates but as recognition accuracy reaches 100%, the different strategies lose their influence on the outcome of the interaction. As long as recognizers remain less than perfect, however, the combination of user instruction and error recovery need to be studied to provide more specific guidelines to speech interface design.

6.4.6 Flexibility of Speech Interface Design Toolkits

PureSpeech is currently developing so-called VUI modules as part of a speech interface design toolkit. As outlined in section 0.2 of the *Preface*, these modules are software objects that implement pre-configured steps of an interaction into the speech interface. An evaluation of the possibilities of implementing and maintaining flexibility in modular speech interface building blocks could provide invaluable insight into the design of these modules. While VUI modules are designed to take care of trivial aspects of the implementation of speech interfaces, they should also allow for enough implementational flexibility. Modules that are developed under contemporary insights should be able to incorporate new empirical findings on speech interface design issues like hierarchical utterance complexity and user sensitive prompt adjustment.

6.5 Summary and Conclusion

In an interaction with an ASR application, users have a preference for a level of utterance complexity that allows multiple information specification in a small number of interaction steps. For more complex utterances, it is necessary to provide the user with guidance on how to phrase the input, for instance by providing examples, and on the discourse itself, for example by providing conversational cues familiar from natural human conversation. Large individual differences exist, however, in the need for such guidance. An optimal speech application would therefore adjust prompting techniques to the needs of the user through an interface structure with a hierarchical utterance complexity and user sensitive prompt adjustment. Such a system could monitor and optimally follow the dynamic balance on the speed-accuracy trade-off during the interaction.

Preparatory to developing such a highly flexible and sensitive system, further insights in several related issues need to be accumulated. Besides fine tuning the findings of this study with more advanced speech recognizers, sociolinguistic research, and further study of error recovery and user instruction, investigations in new areas are necessary as well. These areas include flexibility of VUI design toolkits, the combination of different levels of utterance complexity, and variability of prompting. It is hoped that this study will stimulate further investigation, especially in the fields of hierarchical utterance complexity and user sensitive prompt adjustment.

Appendix A: Implementational Choices

Description and justification of the decisions made with respect to the implementation of the experiment application

For the implementation of the experiment application for this study a number of choices had to be made that were not directly related to the experimental design. These implementational choices are described below.

A.1 Four Different Systems

The four systems that were built for the experiment application differed in their domain, but were similar in their functionality. Otherwise they would not have been suitable for comparison. The chosen systems—commuter rail schedule information, auto loan calculator, auto blue book value, and catalog ordering—were all systems providing certain information—as opposed to systems performing a certain action, such as transferring money between bank accounts, for example. To find that information, they all needed four pieces of data from the user.

As outlined in paragraph 4.3.1.1, the simple-utterance interaction consisted of four recognition steps and the complex-utterance interaction of two. Ideally, the two requested pieces of information at both steps in the complex condition would fit together well intuitively, and over the different systems all information pairs would appear equally natural as well. Any imbalance in this respect would introduce a bias in favor of the domain that had better information pairs.

Not all information pairs seemed to be evenly intuitively correct, however. The make and model of a car, for instance, combine intuitively into a very strong pair—such as "Ford Taurus", for example—while the period and interest rate of a car loan may combine less easily—such as "over 3 years at an interest rate of 7.9% percent", for example. System-condition pairing was counterbalanced to eliminate any variance due to nuisance variables, however.

A.2 Call-Flows

The four systems for this experiment were contained in a single application. Callers entered the system through the Call-In Menu (CM)—see also Figure A.1. The function of CM is to greet the subject and to prompt them to enter their access code. The access code consisted of four digits, of which the first two represented the subject number. The last two digits were used to control the assignment of experimental conditions to subjects—see also *Appendix E: Subjects and Access Codes Overview*. Depending on the third digit, the access code led to the Main Menu (MM)—see also Figure A.2—of one of the four systems. The function of MM was to greet the caller, tell them what system they had reached, inform them that the system used ASR—which was in concurrence with the Schumacher *et al.* (1995) guideline that users should be made aware that they are about to talk to a speech recognizer—and to give them instructions on how to use the system. From there on, the system led to one of the eight experimental conditions, depending on the last digit of the access code, and it would prompt for the first input.

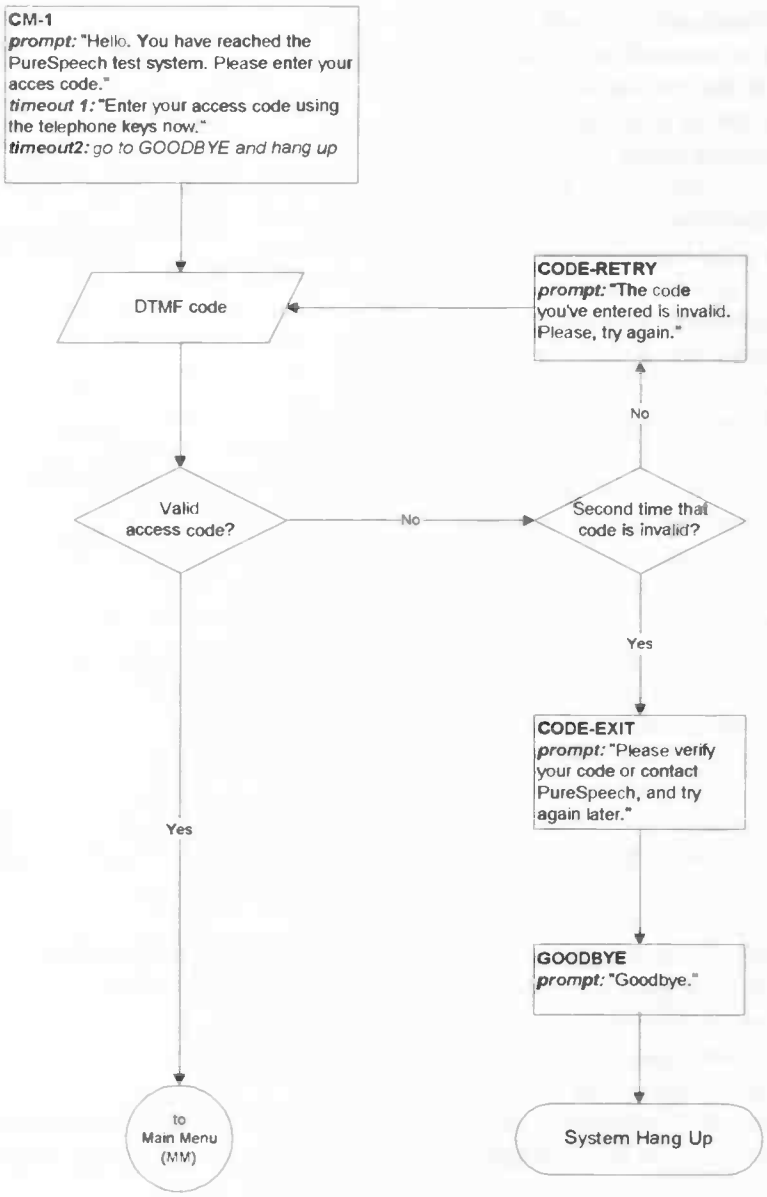


Figure A.1—Call-flow for the Call-In Menu (CM)

A.2.1 Error Recovery

For two reasons, the application needed to have an error recovery mechanism. Such a mechanism would not only give callers the opportunity to rephrase their input in case they were mistaken and had given the wrong input, but it would also give them a chance to reenter their input, if the system had mis-understood what they said.

Ideally the latter case would not occur during the experiments. Since misrecognition was not part of the experiment, it was best if the system always correctly recognized user input. With the speech recognizer that was available for the implementation of the ASR system for this experiment, however, that ideal situation was not likely to occur consistently. To achieve acceptable levels of task compliance despite the limitations of the recognizer, an error recovery mechanism was important.

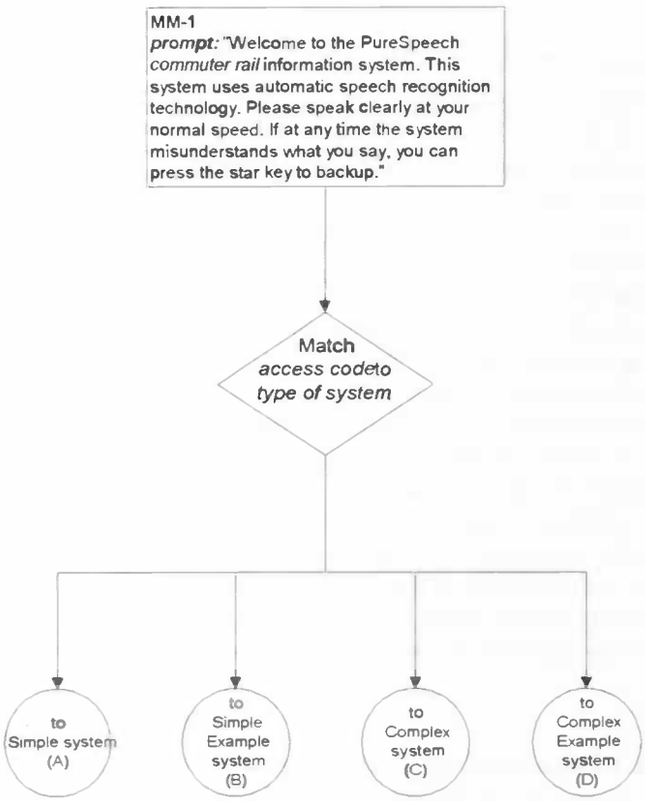


Figure A.2—Call-flow for the Main Menu (MM) of the commuter rail schedule information system

Recovery mechanisms in general should be unobtrusive in an interaction, especially when the rate of information entry is high, as in an IVR information system. This means that they should be rapid and easy to use.

A possible implementation for such a mechanism would be to add a natural language phrase such as "No, that is wrong" to every recognition context. However, this would enlarge recognition contexts and thereby increase the chances of misrecognition, especially for a recognizer that is not very powerful—like the recognizer that was used for this application. In addition, the correction phrase itself may get misrecognized. A possible alternative would be to designate a special DTMF key as an option for the caller to backup, in order to re-enter the input. Such a key is easy to use, and acts directly. Providing feedback to the user in combination with such an option to cancel out an error is necessary when explicit confirmation is not desired while the cost of errors is high (Kamm, 1994).

Casali *et al.* (1990) described a trade-off between costs of high recognition accuracy and necessary time for error recovery. Keeping error recovery times as short as possible by the use of an efficient error recovery mechanism is a way to optimize the balance in the trade-off described by Casali *et al.* (1990). For the experiment application of the study described in this report, the star key was a very quick and straightforward recovery mechanism.

A.2.2 Feedback

Feedback was provided immediately after user input and directly before a new request for user input. With this setup, the user was able to decide to press the star key shortly after giving the input, without having to listen to the entire prompt first. Listening to the next request for information would be a waste of time and cognitive effort, since that system output is unnecessary in case of misrecognition, or an input mistake. If the user had said "Sunday" in the previous state and the system had misrecognized it as "Monday", the second prompt type would have prepared the user to phrase the departure time, until they realized at the end of the prompt that "Sunday" had been misrecognized. The subject would then suddenly have had to change a perhaps already initiated action, and press the star key instead of saying the desired departure time.

A.3 Grammar

A.3.1 Grammar Size and Recognition Accuracy

In order to make it possible for the subjects to complete the scenarios, the grammar specifications had to cover all the data included in the scenarios. In order not to restrict recognition to correct input only, the recognition contexts needed to cover a wider range of user input. Larger contexts make recognition more difficult however, consequently lowering recognition accuracy. As long as the recognizer is robust this will not be a problem, but due to technical circumstances the recognition for the application for this experiment did not have a margin that made large contexts possible.

Low recognition accuracy would influence the results of the experiment. That task completion time is longer and user accuracy is lower for lower levels of recognition accuracy was assessed in an experiment by Casali *et al.* (1990) in which subjects performed a data-entry task via a simulated speech recognizer. They also found that having to interrupt the task to correct errors was annoying and frustrating to the users. Their experiments failed to support the intuitive assumption that accuracy level and vocabulary size of a recognizer would be the primary determinants of user satisfaction. They concluded, however, that when a high information entry level is critical, recognition accuracy levels should be compromised as little as possible. The tasks involved in the experiment described in this report were obviously information-entry and retrieval tasks. Therefore high recognition accuracy was essential.

A.4 Prompts

The prompts used in the experiment can be found in *Appendix C: Prompts Overview* on pages 63-66. The feedback in the prompts in this overview is represented by variable names.

A.4.1 Design and Recording

An essential choice that had to be made for the implementation of the experiment application was whether to use explicit or implicit prompting[26]. This choice would have an influence on the naturalness of the system, and on the likelihood that subjects would adjust the formulation of their input to the style of the system output.

Explicit prompts are much more directive, as can be seen in the following example for the first step in the simple utterance condition of the commuter rail system:

> *Explicit prompting:* "Say the name of the station you would like to leave from."
> *Implicit prompting:* "Which station would you like to leave from?"

Explicit prompts are especially useful for guiding an interaction when restricted input is required by the system, since they increase user compliance to the restrictions (Kamm, 1994). The subjects for this experiment had to restrict their input to either simple phrases or more complex phrases. Despite Kamm's (1994) argument, however, implicit prompting was implemented for the application for this experiment, because this gave the interface a more natural appearance than if explicit prompts had been used.

The first prompt of every state was *implicitly directive* through the use of an implicit template inducing the subject to model their input accordingly. Ideally, this way the

---

[26] *Modeling* and *shaping*, as described by Zoltan-Ford (1989), refer to implicit and explicit prompting respectively. Please see paragraph 3.4, *Prompting and User Input Curtailment*, for an elaboration on modeling and shaping.

subjects would automatically adjust their phrases to the limitations of the application, without realizing it and without noticing the limitations, while the system would appear very natural. In inter-human communication, people frequently model the style and content of what they say after what is said by their conversational partner, and they do this without being aware of it (Danzinger, 1976; as referred to by Zoltan-Ford, 1991).

Directive prompts are suitable for following up errors (Kamm, 1994). Explicit prompting is more effective in reducing the variability of speech input than implicit prompting (Zoltan-Ford, 1991). Explicitly directive prompts were chosen for time-out and for error prompts to provide the extra guidance necessary. They were explicit requests shaping the input of the subjects.

People rely on IVR systems to be at service with attention, capability and authority, and therefore prefer machines to be somewhat assertive (Buchheit *et al.*, 1990). Accordingly, prompts were kept to the point and as brief as possible. Also, they were phrased and spoken in a polite manner and a friendly tone, were unhurried and of low-pressure. That assertiveness can be combined with friendliness is indicated by people's appreciation of the use of "please" and "thank you" in ASR application output (Zoltan-Ford, 1991).

A.4.2 Feedback

Feedback can be provided either by explicit or implicit confirmation. The difference between the two options is similar to explicit and implicit prompting, as the following comparison demonstrates:

Explicit feedback: "Did you say 'North Station'?"
Implicit feedback: "Okay, from 'North Station'."

With explicit feedback an extra recognition step is necessary for each confirmation. As soon as the user responds to this feedback with an affirmative answer, the system continues to the next state of the interaction. This extra step is not necessary with implicit confirmation, since this type of feedback will be provided as part of the prompting in the following state. As long as the user does not initiate some kind of error recovery, the interaction simply continues.

Explicit confirmation or implicit feedback combined with a backup facility is necessary when the costs of misrecognition are high (Kamm, 1994). In an interaction where the user provides the system with the necessary data, costs of misrecognition are high because they would lead to an incorrect end-result. Feedback has therefore to be provided in those situations.

To make sure the interface appeared as natural as possible, implicit confirmation was implemented in the application for our experiment. A natural interface was better suited for this experiment, because an interface that appears more mechanical could perhaps influence the way users spoke to the system. Also, this was consistent with the goal of the overall NEI project to develop a transit and way-finding information system prototype that is as friendly and natural as possible. The costs of misrecognition were kept to a minimum as well by providing feedback immediately after the input and the star key of the telephone key-pad as a fast and easy to use backup mechanism.

In the states where the application reported to the user the information requested, the costs of misrecognition on further options were minimal. Therefore, the input in those contexts was acted upon without further confirmation.

A.4.3 Examples

For all pieces of input data the subjects would hear more than one example in the example condition if a time-out prompt or rejection prompt was played by the system.

For certain types of information input there were even more examples, because some different states required the same type of input—station names, for instance—and because certain other states required very similar input—such as price of the car, compared to down payment or mileage of the car.

The exact values given in the example prompts differed from the values used in the scenarios so the subjects could not just literally repeat the examples when interacting with the system. Also, subjects were not given the same example in different prompts, because this could lead to confusion and mistakes making the outcome of the experiment unreliable.

Another reason why examples differed was to make the callers aware of the variety of the system. This was done by choosing values for the examples that differed in as many aspects from each other as possible. For station names, this would mean that the names given as examples consisted of a single word or two words, corresponded to stations on different sides of the commuter rail line, were either long or short, etc. For times, it was made sure that there would be examples with "AM" as well as with "PM". For cars, there were examples with American-made cars as well as non-American cars. For every type of information the entire range of its variety was represented by the examples.

As described in paragraph 4.4.3.3, *Examples*, since every subject would go through one different experimental condition for each of the four systems, examples were kept fully consistent over the complex versus the simple condition. Consistency was maintained over the actual data used for the examples, and the type of prompt they appeared in. This consistency was necessary to avoid differences between the various conditions that were not part of the experimental treatments or could introduce a certain bias.

A.5 Scenarios

There was a possibility that a subject would not succeed on a task because of technical instabilities of the system or because of task difficulty. If there had been only one scenario per system and a user failed to accomplish the task for a scenario, then data would not be available for the combination of a particular domain and the respective experimental condition. This would then leave an empty spot in the data for that particular system and condition. Therefore, two scenarios were written for each of the four systems—see also *Appendix D: Scenario Overview*.

Thus, if one scenario had been unsuccessful, the data from the second scenario for the same domain was still available, unless a subject failed on both scenarios for the system. For every system-condition combination the data was used from both scenarios when available, or from either one of the two scenarios otherwise.

Appendix B: Recruitment Poster

Want $20?

PURESPEECH™
THE NATURAL INTERFACE

PureSpeech, Inc. will give you twenty dollars for an hour of your time, to help us test some new telephone information systems.

To qualify, you must be at least 18 years old and you must have used an automated telephone information system (like telephone banking) in the last 12 months.

Interested?
Call us at PureSpeech, Inc. at 876-4079

PURESPEECH™	PURESPEECH™	PURESPEECH™	PURESPEECH™	PURESPEECH™
testing new telephone information systems	testing new telephone information systems	testing new telephone information systems	testing new telephone information systems	testing new telephone information systems
876-4079	876-4079	876-4079	876-4079	876-4079

Appendix C: Prompts Overview

The 'prompts' are the first prompts played in every step, and 'time-outs' are the time-out prompts played when no input was detected. Feedback, which was dependent on user input, is indicated by variables starting with a '%' sign.

C.1 Commuter Rail System

	Simple, No Example:	Simple, Example:
Step 1	**prompt**: "Which station would you like to leave from?" **time-out**: "Please say the name of your departure station."	**prompt**: "Which station would you like to leave from? For example: 'Prides Crossing'." **time-out**: "Please say the name of the departure station. For example: 'Ipswich'."
Step 2	**prompt**: "Leaving *%dep_stat*. Which station would you like to arrive at?" **time-out**: "Please say the name of your arrival station, or press 'star' to back up."	**prompt**: "Leaving *%dep_stat*. Which station would you like to arrive at? For example: 'Salem'." **time-out**: "Please say the name of the arrival station. For example: 'Chelsea', or press 'star' to back up."
Step 3	**prompt**: "Okay, from *%dep_stat* to *%arr_stat*. On what day of the week?" **time-out**: "Please say the day of the week you'd like to travel on, or press 'star' to back up."	**prompt**: "Okay, from *%dep_stat* to *%arr_stat*. On what day of the week? For example: Wednesday." **time-out**: "Please say the day of the week you'd like to travel on, for example: Saturday, or press 'star' to back up."
Step 4	**prompt**: "On *%day*. Around what time, to the nearest half hour, would you like to depart?" **time-out**: "Please say the time, to the nearest half hour, you'd like to depart, or press 'star' to back up."	**prompt**: "On *%day*. Around what time, to the nearest half hour, would you like to leave? For example: 3 PM." **time-out**: "Please say the time, to the nearest half hour, you want to leave, for example: 11 AM. Or press 'star' to back up."

	Complex, No Example	Complex, Example:
Step 1	**prompt**: "From which station to which station would you like to travel?" **time-out**: "Please say from which station to which station you want to travel."	**prompt**: "From which station to which station would you like to travel? For example: from Prides Crossing to Salem." **time-out**: "Please say from which station to which station you want to travel. For example: from Ipswich to Chelsea."
Step 2	**prompt**: "Okay, from *%dep_stat* to *%arr_stat*. Around what time, to the nearest half hour, and on what day of the week would you like to leave ?" **time-out**: "Please say around what time, to the nearest half hour, and on what day of the week you'd like to leave, or press 'star' to back up."	**prompt**: "Okay, from *%dep_stat* to *%arr_stat*. Around what time, to the nearest half hour, and on what day of the week would you like to leave? For example: 3PM on Wednesday." **time-out**: "Please say around what time, to the nearest half hour, and on what day of the week you'd like to leave. For example: 11AM on Saturday. Or press 'star' to back up."

C.2 Auto Loan Calculator

	Simple, No Example:	***Simple, Example:***
Step 1	**prompt**: "What is the price of the car?" **time-out**: "Please say the price of the car."	**prompt**: "What is the price of the car? For example: 15,000 dollars." **time-out**: "Please say the price of the car. For example: 32,500 dollars."
Step 2	**prompt**: "What is your down payment?" **time-out**: "Please say the amount of your down payment. Or press 'star' to back up."	**prompt**: "What is your down payment? For example: 2,000 dollars." **time-out**: "Please say the amount of your down payment. For example: 4,500 dollars, or press 'star' to back up."
Step 3	**prompt**: "Okay, *%amount* dollar loan, with *%down* dollars down. What is the annual interest rate?" **time-out**: "Please say the amount of the annual interest rate, or press 'star' to back up."	**prompt**: "Okay, *%amount* dollar loan, with *%down* dollars down. What is the annual interest rate? For example: 6.7%" **time-out**: "Please say the annual interest rate. For example: 5%, or press 'star' to back up."
Step 4	**prompt**: "*%interest* percent. Over how many years?" **time-out**: "Please say over how many years the loan will be paid, or press 'star' to back up."	**prompt**: "*%interest* percent. Over how many years? For example: 5 years." **time-out**: "Please say over how many years the loan will be paid, like for example: 8 years, or press 'star' to back up."
	Complex, No Example	***Complex, Example:***
Step 1	**prompt**: "What is the price of the car, and what is your down payment?" **time-out**: "Please say the price of the car and the amount of your down payment."	**prompt**: "What is the price of the car, and what is your down payment? For example: 15,000 dollars with 2,000 dollars down." **time-out**: "Please say the price of the car and your down payment, for example: 32,500 dollars with 4,500 dollars down."
Step 2	**prompt**: "Okay, *%amount* dollars with *%down* dollars down. Over how many years, at what interest rate?" **time-out**: "Please say over how many years the loan will be paid, and at what interest rate. Or press 'star' to back up."	**prompt**: "Okay, *%amount* dollars with *%down* dollars down. Over how many years, at what interest rate? For example: over 5 years, with 6.7% interest." **time-out**: "Please say over how many years the loan will be paid and at what interest rate, for example: over 8 years, with 5% interest. Or press 'star' to back up."

C.3 Auto Blue Book

	Simple, No Example:	***Simple, Example:***
Step 1	**prompt**: "What is the make of the car?" **time-out**: "Say the make of the car."	**prompt**: "What is the make of the car? For example: 'Ford'." **Time-out**: "Please say the make of the car, for example: 'Honda'."
Step 2	**prompt**: "What model *%make*?" **time-out**: "Please say the model of the *%make* you're interested in. Or press 'star' to back up."	**prompt**: "What model *%make*? For example: *%model*" **time-out**: "Please say the model of the *%make* you're interested in, for example: *%model*. Or press 'star' to back up."
Step 3	**prompt**: "What year is the *%make %model*?" **time-out**: "Please say the year of the *%make %model*. Or press 'star' to back up."	**prompt**: "What year is the *%make %model*? For example: 1990." **time-out**: "Please say the year of the *%make %model*, for example: 1995. Or press 'star' to back up."
Step 4	**prompt**: "From *%year*. What is its mileage?" **time-out**: "Please say the mileage of the *%year %make %model*. Or press 'star' to back up."	**Prompt**: "Okay *%year*. What is its mileage? For example: 24,000 miles." **time-out**: "Please say the mileage of the *%year %make %model*, for example: 10,000 miles. Or press 'star' to back up."
	Complex, No Example	***Complex, Example:***
Step 1	**prompt**: "What is the make and model of the car?" **time-out**: "Please say the make and model of the car."	**Prompt**: "What is the make and model of the car? For example: 'Ford Taurus'." **time-out**: "Please say the make and model of the car, for example: 'Honda Accord'."
Step 2	**prompt**: "Okay, *%make %model*. What year is the car, and what is its mileage?" **time-out**: "Please say the year of the car and say its mileage. Or press 'star' to back up."	**prompt**: "Okay, *%make %model*. What year is the car, and what is its mileage? For example: from 1990, with 24,000 miles." **time-out**: "Please say the year of the car, and say its mileage, for example: from 1995, with 10,000 miles. Or press 'star' to back up."

C.4 Catalog Order System

	Simple, No Example:	***Simple, Example:***
Step 1	**prompt**: "What item are you interested in?" **time-out**: "Please say the name of the item you're interested in."	**prompt**: "What item are you interested in? For example: T-shirt." **time-out**: "Please say the name of the item you're interested in, for example: turtleneck."
Step 2	**prompt**: "How many *%item(s)* would you like to purchase?" **time-out**: "Please say the number of *%item(s)* you'd like to purchase, or press 'star' to back up."	**prompt**: "How many *%item(s)* would you like to purchase? For example: 1." **time-out**: "Please say the number of *%item(s)* you'd like to purchase, for example: 3, or press 'star' to back up."
Step 3	**prompt**: "In what size would you like the *%quantity %item(s)*?" **time-out**: "Please say the size of the item, or press 'star' to back up."	**prompt**: "In what size would you like the *%quantity %item(s)*? For example: medium." **time-out**: "Please say the size of the item, for example: extra large, or press 'star' to back up."
Step 4	**prompt**: "Okay, *%size*. What color would you like?" **time-out**: "Please say the color of the item, or press 'star' to back up."	**prompt**: "Okay, *%size*. What color would you like? For example: red." **time-out**: "Please say the color of the item, for example: gray, or press 'star' to back up."
	Complex, No Example	***Complex, Example:***
Step 1	**prompt**: "What is the quantity and name of the item you're interested in?" **time-out**: "Please say the quantity and name of the item."	**prompt**: "What is the quantity and name of the item you're interested in? For example: 1 T-shirt." **time-out**: "Please say the quantity and name of the item, for example: 3 turtlenecks."
Step 2	**prompt**: "Okay, *%quantity %item*. What size and color would you like?" **time-out**: "Please say the size and color of the item, or press 'star' to back up."	**prompt**: "Okay, *%quantity %item*. What size and color would you like? For example: medium, red." **time-out**: "Please say the size and color of the item, for example: extra large, gray, or press 'star' to back up."

Appendix D: Scenario Overview

	Scenario 1	Scenario 2
Commuter Rail words/chars: sc.1: 95/386 sc.2: 64/275	Imagine you're starting a new job in Boston next *Monday*. You live near the *Rockport* commuter rail station, so you figure that taking the commuter rail to work would be a great way to get there. You look at your map and see that *North Station* is next to where you work. Because you want to make a great first impression, you think getting to work on time would be a good thing. Call the Commuter Rail System to find out what time you get to work if you take a train around *7:30 am*.	Imagine that you are at work on a *Saturday* that just happens to be Halloween. Some friends want to go to *Salem* after work to take part in the Halloween festivities. You get off of work at *5:00 p.m.* Assuming that you catch the next train leaving *North Station*, call the Commuter Rail System to find out what time you would arrive in Salem.
Auto Loan words/chars: sc.1: 94/396 sc.2: 84/361	Imagine that you are in the process of buying a new car. You decide to buy a Volkswagon Jetta. With the options you need, the price of the car comes to *$17,000*. You can scrape together *$1,000* to use as a down payment, but you will need to borrow the rest. Your bank is offering a *3 year* auto loan with an interest rate of *7.9%*. They also provide a telephone based system that tells you your monthly payments. Call the Auto Loan Calculator to find out the monthly payment of this auto loan.	Imagine that you are currently paying off a car loan. There is *$12,000* left to pay on the loan at a 9.5% annual interest rate. Your monthly payments are $352. You recently heard that your bank is offering personal loans at a special rate of *7.1%*. You also have *$2,000* cash that you could apply to the loan. Call the Auto Loan Calculator to find out what your new monthly payments would be if you refinanced your car with a *4 year* personal loan.
Blue Book words/chars: sc.1: 63/250 sc.2: 70/284	Imagine that you currently own a *1992 Ford Taurus* with *52,000 miles*. You decide that you're going to buy a new car and trade in your old one. To make sure that you are treated fairly at the dealer, you want to find out the value of your old car. Call the Auto Blue Book System to determine the value of your car.	Imagine that a friend of yours is interested in buying a used *Toyota Camry*. She has her eyes on a *1994* model with *41,000 miles*. The owner is asking $19,000 for it, but she isn't sure if that is a fair price. You offer to find out the car's blue book value. Call the Auto Blue Book System to find out whether the amount the owner is asking is reasonable.
Catalog Order words/chars: sc.1: 51/221 sc.2: 46/226	Imagine that you hate going to the mall. You need some clothes, however, so you decide to purchase some clothing by mail order. You really like *sweaters* and you want to add a *large blue* one to your collection. Call the Catalog Order System to find out how much *one* costs.	Imagine that you are a grandmother who always buys *green T-shirts* for her grandchildren on their birthdays. *Two* of your grandchildren have birthdays coming up. Both of the youngsters are size *medium*. Call the Catalog Order System to find out how much your gift will cost.

Appendix E: Subjects and Access Codes Overview

1 = Commuter Rail System
2 = Auto Loan Calculator
3 = Auto Blue Book System
4 = Catalog Order System

A = simple
B = simple/example
C = complex
D = complex/example

Without beep tones:

	subj.#		Latin square	access codes	date	time
☐	01	♡	1a/2b/3c/4d	0111 0122 0133 0144	1997	.....
☐	02	♡	1b/2c/3d/4a	0212 0223 0234 0241	1997	.....
☐	03	♡	1c/2d/3a/4b	0313 0324 0331 0342	1997	.....
☐	04	♡	1d/2a/3b/4c	0414 0421 0432 0443	1997	.....
☐	05	♠	2a/3b/4c/1d	0521 0532 0543 0514	1997	.....
☐	06	♠	2b/3c/4d/1a	0622 0633 0644 0611	1997	.....
☐	07	♠	2c/3d/4a/1b	0723 0734 0741 0712	1997	.....
☐	08	♠	2d/3a/4b/1c	0824 0831 0842 0813	1997	.....
☐	09	♣	3a/4b/1c/2d	0931 0942 0913 0924	1997	.....
☐	10	♣	3b/4c/1d/2a	1032 1043 1014 1021	1997	.....
☐	11	♣	3c/4d/1a/2b	1133 1144 1111 1122	1997	.....
☐	12	♣	3d/4a/1b/2c	1234 1241 1212 1223	1997	.....
☐	13	❖	4a/1b/2c/3d	1341 1312 1323 1334	1997	.....
☐	14	❖	4b/1c/2d/3a	1442 1413 1424 1431	1997	.....
☐	15	❖	4c/1d/2a/3b	1543 1514 1521 1532	1997	.....
☐	16	❖	4d/1a/2b/3c	1644 1611 1622 1633	1997	.....

With beep tones:

	subj.#		Latin square	access codes	date	time
☐	17	♡	1a/2b/3c/4d	1715 1726 1737 1748	1997	.....
☐	18	♡	1b/2c/3d/4a	1816 1827 1838 1845	1997	.....
☐	19	♡	1c/2d/3a/4b	1917 1928 1935 1946	1997	.....
☐	20	♡	1d/2a/3b/4c	2018 2025 2036 2047	1997	.....
☐	21	♠	2a/3b/4c/1d	2125 2136 2147 2118	1997	.....
☐	22	♠	2b/3c/4d/1a	2226 2237 2248 2215	1997	.....
☐	23	♠	2c/3d/4a/1b	2327 2338 2345 2316	1997	.....
☐	24	♠	2d/3a/4b/1c	2428 2435 2446 2417	1997	.....
☐	25	♣	3a/4b/1c/2d	2535 2546 2517 2528	1997	.....
☐	26	♣	3b/4c/1d/2a	2636 2647 2618 2625	1997	.....
☐	27	♣	3c/4d/1a/2b	2737 2748 2715 2726	1997	.....
☐	28	♣	3d/4a/1b/2c	2838 2845 2816 2827	1997	.....
☐	29	❖	4a/1b/2c/3d	2945 2916 2927 2938	1997	.....
☐	30	❖	4b/1c/2d/3a	3046 3017 3028 3035	1997	.....
☐	31	❖	4c/1d/2a/3b	3147 3118 3125 3136	1997	.....
☐	32	❖	4d/1a/2b/3c	3248 3215 3226 3237	1997	.....

Appendix F: Subject Instruction

Prior to the experiment, each subject was given the same basic instructions.

"Welcome and thank-you for participating in our study. As you probably know, PureSpeech produces automatic speech recognition systems that are used to provide information services over the phone. We appreciate your help in testing designs for these systems.

For the next hour we will ask you to read several scenario descriptions contained in a booklet and then call up a speech recognition system to obtain some information or service. We would like you to read the scenarios carefully and try to imagine yourself in the described situation. For instance, if the scenario describes that you want to transfer money from your checking to savings account using a telephone banking system, try to really imagine yourself at home wanting to do this. Also, consider how important it is that the system does exactly what you want it to do.

After reading the scenario, call up the system to find out the information described in the scenario. Punch in the four digit access code given in the booklet and then talk to the system in a clear, natural manner. As you talk to it, the system will occasionally repeat what it thinks you said. Please pay careful attention to this information and to the instructions the system will give you on how to handle misrecognitions. Occasionally, the system may not understand what you said to it or it may think you said something different than you really said. In these cases, try to back up and get the system to understand what you want. If it persists in mis-understanding you, try several more times. If you are not sure what to do at any time while you are talking to the system, just use your best judgment. Keep in mind that this is a test system that is still under development. It is important for us to find out what kinds of trouble the system has, and what kinds of difficulties people have using it. Your participation in this study will help us to make the system work better.

When you have the information needed in each scenario, you may simply hang up. Then turn to the next page in your booklet. You will see a page with several lines on it and descriptive words at either end of the lines. For each pair of words, think about how you would rate the system with respect to those descriptions and place a mark at the location on the line that you feel represents the system's performance. Each scenario you will complete will be designed to function a little differently, so think carefully about the scenario you have just completed each time you fill out this sheet. After completing all the scenarios, please feel free to write any specific comments you have about the system on the back of the sheet.

Once again, thank-you for participating and we hope you have a little fun with this study. Please feel free to ask me any questions you have about the study at any time."

Appendix G: Experiment Work Sheet

Please dial in to the telephone information system, and use the access codes listed below, and fill in the answers to the scenarios.

scenario	*access code*	*answer*
Commuter Rail System 1	› ›	train number.: ..... arrival time: .....
Commuter Rail System 2	› ›	train number.: ..... arrival time: .....
Auto Loan Calculator 1	› ›	monthly payment: .....
Auto Loan Calculator 2	› ›	new monthly payment: .....
Auto Blue Book System 1	› ›	value of the car: .....
Auto Blue Book System 2	› ›	value of the car: .....
Catalog Order System 1	› ›	price of the order: .....
Catalog Order System 2	› ›	price of the order: .....

Appendix H: Bipolar Rating Scales

Please rate the system you just used on the dimensions listed below by placing a pencil mark at the position on the line that most accurately reflects your impression of the system.

Unacceptable	_____	Acceptable
Slow	_____	Fast
Inaccurate	_____	Accurate
Inconsistent	_____	Consistent
Irritating	_____	Pleasing
Undependable	_____	Dependable
Unnatural	_____	Natural
Incomplete	_____	Complete
Uncomfortable	_____	Comfortable
Unfriendly	_____	Friendly
Distracting	_____	Facilitating
Complicated	_____	Simple
Useless	_____	Useful

Appendix I: Experiment System Preference Form

For each of the following systems, place its letter on the line below, to indicate how much you liked or disliked using it.

R - Commuter Rail System

B - Blue Book Calculator

L - Loan Calculator

O - Catalog Ordering System

dislike _____ like

Feel free to write down some additional comments about the systems you have used in this experiment :

(you can continue on the other side if you like)

References

Indirect references are marked with '‡'.

Ainsworth, W.A. (1993)—Technical note: theoretical and simulation approaches to error correction strategies in automatic speech recognition. *International Journal of Man-Machine Studies, 39, pp. 517 - 520.*

Aust, Harald, Martin Oerder, Frank Seide & Volker Steinbiss (1994)—The Philips automatic train timetable information system. *IEEE Proceedings, Second Workshop on Interactive Voice Technology for Telecommunications Applications.*

Aust, Harald, & Martin Oerder (1995)—Dialogue control in automatic inquiry systems. *ESCA Workshop on Spoken Dialogue Systems, Vigso, Denmark, pp. 121 - 124.*

Baber, C., & K.S. Hone (1993)—Modeling error recovery and repair in automatic speech recognition. *International Journal of Man-Machine Studies, 39, pp. 495 - 515.*

Blasch‡, B., & L. Hiatt (1983)—Orientation and wayfinding. *U.S. Architectural and Transportation Barriers Compliance Board Technical Paper.*

Biermann, Alan W., Linda Fineman & J. Francis Heidlage (1992)—A voice- and touch-driven natural language editor and its performance. *International Journal of Man-Machine Studies, 37, pp. 1 - 21.*

Billi, Roberto, Castagneri Guiseppe & Morena Danieli (1996)—Field trial evaluations of two different information inquiry systems. *IEEE Proceedings, Third Workshop on Interactive Voice Technology for Telecommunications Applications, pp. 129 - 134, Basking Ridge, NJ.*

Brems, Douglas J., Michael D. Rabin & Jill L. Waggett (1995)—Using natural language conventions in the user interface design of automatic speech recognition systems. *Human Factors, 37(2), pp. 265 - 282.*

Brooks‡, C.M. (1989)—User interface design for speech recognition telephone applications. *Speech Technology, pp. 58 - 60.*

Buchheit, Paul & Thomas Moher (1990)—Response assertiveness in human-computer dialogues. *International Journal of Man-Machine Studies, 32, pp. 109 - 117.*

Bunt‡, H.C. (1988)—On-line interpretation in speech understanding and dialogue systems. *In H. Niemann, M. Lang & G. Sagerer, Eds., Proceedings of the NATO ASI: Recent Advances in Speech Understanding and Dialog Systems, Berlin: Springer, pp. 349 - 395.*

Capindale, Ruth A. & Robert G. Crawford (1990)—Using a natural language interface with casual users. *International Journal of Man-Machine Studies, 32, pp. 341 - 362.*

Casali, Sherry P., Beverly H. Williges & Robert D. Dryden (1990)—Effects of recognition accuracy and vocabulary size of a speech recognition system on task performance and user acceptance. *Human Factors, 32(2), pp. 183 - 196.*

Chigier, Benjamin (1996)—NEI SBIR phase II proposal. *Unpublished research proposal for phase II of the NEI SBIR, pp. 1 - 45, PureSpeech, Inc., Cambridge, MA.*

Coleman‡, W.D. (1985)—Examining the relationship between performance measures and user evaluations in a transfer-of-training paradigm. *Unpublished Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA.*

Damper, R. I., & Wood, S. D. (1995)—Speech versus keying in command and control applications. *International Journal of Man-Machine Studies, 42, pp. 289 - 305.*

Danzinger‡, K. (1976)—Interpersonal communication. *New York: Pergamon Press.*

Dix, Alan, Janet Finlay, Gregory Abowd & Russell Beale (1993)—Human-Computer Interaction. *Prentice Hall, Hertfordshire, UK.*

Engel, Frits L., & Reinder Haakma (1993)—Expectations and feedback in user-system communication. *International Journal of Man-Machine Studies, 39, pp. 427 - 452.*

Fay‡, D. (1992)—Touch-tone or automated automatic speech recognition: Which do people prefer? *In AVIOS '92 Proceedings—Voice I/O Systems Applications Conference, Minneapolis, MN: American Voice I/O Society, pp. 207 - 213.*

Kamm, Candace (1994)—User interfaces for voice applications. *Voice communication between humans and machines, National Academy Press, Washington, DC, pp. 422 - 442.*

Karis, Demetrios & Kathryn M. Dobroth (1991)—Automating services with speech recognition over the public switched telephone network. *Human Factors Considerations in IEEE Journal on Selected Areas in Communication, 9(4), pp. 574 - 585.*

Martin, Gale L. (1989)—The utility of speech input in user-computer interfaces. *International Journal of Man-Machine Studies, 30, pp. 355 - 375.*

Lea‡, W.A. (1983)—Problems in predicting the performance of speech recognizers. *In D. Pallet (Ed.), Proceedings of the Workshop on Standardization for Speech I/O Technology, pp. 15 - 24, Gaithersburg, MD: National Bureau of Standards.*

Leiser‡, R.G. (1989)—Improving natural language and speech interfaces by the use of metalinguistic phenomena. *Applied Ergonomics, 20, pp. 168 - 173.*

Luce‡, P., T. Fuestel & D. Pisoni (1983)—Capacity demands in short term memory for synthetic and natural speech. *Human Factors, 25, pp. 17 - 32.*

Nye‡, J.M. (1982)—Human factors analysis of speech recognition systems. *Speech Technology, 1, pp. 50 - 57.*

Paris, Carol R., Richard D. Gilson, Margaret H. Thomas & Clayton Silver (1995)—Effect of synthetic voice intelligibility on speech comprehension. *Human Factors, 37(2), pp. 335 - 340.*

Poock‡, G.K. (1980)—Experiments with voice input for command and control: using voice input to operate a distributed computer network. *Naval Postgraduate School Report, NPS55-80-016, Monterey, CA.*

Ralston‡, J.V., D.B. Pisoni, S.E. Lively, B.G. Greene & J.W. Mullennix (1991)—Comprehension of synthetic speech produced by rule: Word monitoring and sentence-by-sentence listening times. *Human Factors, 33, pp. 471 - 491.*

Schumacher, Robert M., Mary L. Hardzinski & Amy L. Schwartz (1995)—Increasing the usability of interactive voice response systems: research and guidelines for phone-based interfaces. *Human Factors, 37(2), pp. 251 - 264.*

Shallice‡, T., P. McLeod & K. Lewis (1985)—Isolating cognition modules with the dual-task paradigm: Are speech perception and production modules separate? *Quarterly Journal of Experimental Psychology, 37, pp. 507 - 532.*

Spiegel‡, M.F., M.J. Altom, M.J. Macchi & K.L. Wallace (1988)—Using a monosyllabic test corpus to evaluation of the intelligibility of synthesized and natural speech. *Proceedings of the American Voice I/O Society, San Jose, California.*

Stifelman, Lisa J., Barry Arons, Chris Schmandt & Eric A. Hulteen (1993)—VoiceNotes: a speech interface for a hand-held voice notetaker. *INTERCHI'93 Proceedings, ACM, pp. 179 - 186.*

Wickens, Christopher D. (1992)—Engineering Psychology and Human Performance. *HarperCollins Publishers, Inc., New York, NY.*

Yankelovich, Nicole, Gina-Anne Levow & Matt Marx (1995)—Designing SpeechActs: issues in speech interfaces. *CHI'95 Proceedings, Conference on Human Factors in Computing Systems, Denver, CO.*

Yankelovich, Nicole (1996)—How do users know what to say? *Sun Microsystems Laboratories.*

Yankelovich, Nicole (1996)—Using natural dialogs as the basis for speech interface design. *Sun Microsystems Laboratories.*

Zoltan-Ford, Elizabeth (1991)—How to get people to say and type what computers can understand. *International Journal of Man-Machine Studies, 34, pp. 527 - 547.*

Index

access code, 32; 36; 56; 68; 69; 70; 71
Ainsworth, 55; 74
answer sheet, 32
ASM, 37; 39
Aust, *et al.*, 17; 74
Baber, *et al.*, 55; 74
backup mechanism, 12; 60
barge-through, 13; 28; 34; 52; 53; 54
beep tones, 28; 30; 34
between-subject variable. *See* variable
Biermann, *et al.*, 34; 74
Billi, *et al.*, 29; 74
Blasch, *et al.*, x; 74
blip tones. *See* beep tones
Brems, *et al.*, 8; 54; 74
Brooks, 8; 74
Buchheit, *et al.*, 22; 60; 74
Bunt, 26; 74
call-flow, 13–14; 36–39; 56–58
  call-in menu. *See* CM
  complex interaction. *See* CSM
  design, 13–14
  error recovery, 39; 57–58
  feedback, 39; 58
  main menu. *See* MM
  simple interaction. *See* ASM
Capindale, *et al.*, 7; 10; 19; 20; 28; 29; 34; 74
Casali, *et al.*, 8; 9; 11; 34; 58; 59; 74
Chigier, x; 74
CM, 56; 57; 76
Cognitive Science and Engineering, ix–x
Coleman, 34; 75
computational complexity, 25
confidence. *See* user confidence
confidence measure, 12
confirmation
  explicit, 12; 58; 60
  implicit, 40; 60
consistency, 40; 61
continuous speech, x; 21
conversational cues, 4; 9; 12; 13; 27; 28; 31; 51; 53; 55
cost-effectiveness, iv; x; 2; 8; 16; 24; 25; 31
counterbalancing, 34–35
countermanding, 12
CSM, 38; 40
Damper, *et al.*, 18; 19; 75
Danzinger, 9; 60; 75
data, 43–49
  user acceptance, 47–49
  user performance, 43–47
directory assistance, 25
Dix, *et al.*, 13; 18; 36; 75
DTMF, 1; 3
  tree, 5
Dual Tone Multi Frequency. *See* DTMF
efficiency, iv; 1; 3; 6; 8; 9; 12; 13; 15; 17; 18; 19; 20; 21; 22; 24–25; 27; 31; 54
Engel, *et al.*, 4; 11; 26; 75
error
  correction, 54–55
  recovery, 39; 57–58
  system, 8; 11; 21; 32; 34; 47
  user, 11; 18; 34; 52
explicit confirmation. *See* confirmation
explicitly directive. *See* prompts
extraneous speech, 8; 10; 24; 25; 28; 41
Fay, *et al.*, 14; 75
feedback, 11; 39; 40; 58; 60
grammar, 40–41; 59
hardware, 6; 8; 25
hierarchical utterance complexity, 53–54
Human Factors, ix; x; 4; 8; 9; 15; 20; 51
human resources, 7; 54
hybrid interface structuring, 51
implicit confirmation. *See* confirmation
implicitly directive. *See* prompts
input
  curtailment, 26–27
  examples, 28; 29–30; 34; 40; 60–61
intentions, 8; 11
Interactive Voice Response. *See* IVR
IVR, 1–2
Kamm, *et al.*, 10; 12; 24; 58; 59; 60; 75
Karis, *et al.*, 21; 75
key-pad, 5; 6; 7; 14; 16; 17; 18; 19; 20; 39; 60
Latin Square, 35
layered-protocol model, 11
Lea, 9; 75
Leiser, 24; 75
Luce, *et al.*, 10; 75
Martin, 18; 19; 24; 74; 75
Massachusetts Bay Transportation Authority. *See* MBTA
MBTA, 16; 17; 19; 20
misrecognition, 8; 11; 15; 25; 32; 33; 39; 41; 44; 45; 48; 51; 57; 58; 60; 70
MM, 56; 58
modeling, 23; 26; 27; 30; 59
Multiple-Resource Theory, 7
National Eye Institute. *See* NEI
natural language, x; xi; 1; 3; 4; 6; 7; 8; 10; 11; 14; 19; 20; 21; 22; 23; 24; 51; 54; 55; 58
natural speech. *See* natural language

NEI, x–xi
Nye, 19; 75
NYNEX Telecommunications, x; 9
operator, 1; 5; 8; 16; 17; 21; 25; 29; 49
Paris, *et al.*, 10; 75
Philips Reasearch Laboratories, 17; 74
phone-band, 21; 27
phrase spotting, 8
Poock, 18; 75
prompts, 13; 39–40; 59–61
  design and recording, 39–40; 59–60
  explicitly directive, 39; 60
  feedback, 40; 60
  implicitly directive, 39; 59
  input examples, 40; 60–61
  structure, 40
  tapering, 13
  user sensitive prompt adjustment, 54
PureSpeech, Inc., x; 1; 6; 9; 32; 52; 55
Ralston, *et al.*, 10; 75
rating scales, 34; 47–48
recognition accuracy, 8; 9; 21; 24; 25; 31; 41; 48; 49; 51; 52; 53; 55; 58; 59; 74
recognition context, 4; 10; 21; 24; 25; 39; 40; 41; 52; 54; 58; 59
recognizer, 3; 4; 8; 9; 13; 23; 24; 25; 27; 28; 30; 31; 34; 52; 53; 54; 55; 56; 57; 58; 59; 75
rejection, 12; 25; 39; 40; 41; 43; 44; 45; 60
response efficiency, 19; 20; 24
restricted domain, 10; 19; 20; 22; 29
restricted grammar. *See* restricted domain
restricted input, 59
Schumacher, *et al.*, 3; 14; 21; 56; 75
Shallice, *et al.*, 7; 76
shaping, 23; 26; 27; 39; 59; 60
Sociolinguistics, 22–23; 54
speaker independence, x; 21
speaking cues. *See* conversational cues
speech interface. *See* VUI
speech perception, 7
speech production, 7
speech recognition accuracy. *See* recognition accuracy
speech recognizer. *See* recognizer
Spiegel, *et al.*, 10; 76
Stifelman, *et al.*, 12; 54; 76
synthetic speech. *See* TTS
system error. *See* error
system-driven, 17; 22; 24; 31
task completion time, 45–47
task compliance, iv; 9; 21; 22; 27; 34; 57
telephone key-pad. *See* key-pad
Text To Speech. *See* TTS
time-out, 12; 13; 39; 40; 41; 44; 45; 54; 60; 63; 64; 65; 66
touch-tone. *See* DTMF
transit information, x; 20; 36; 60
TTS, 10
turn-taking, 4; 10; 13; 15; 26; 28; 30; 31; 34; 44; 50; 51; 53
unrestricted speech. *See* natural language
user acceptance, 20–26; 34; 47–49
user confidence, 21; 22; 29
user error. *See* error
user freedom, 4; 16; 17; 21; 23; 24; 25; 26; 31; 53; 54
user input curtailment. *See* input curtailment
user instruction, 54–55
user performance, 20–26; 34; 43–47
user sensitive prompt adjustment, 54
utterance complexity, 27–29; 33–34
  hierarchical, 53–54
variable
  between-subject, iv; 33
  within-subject, iv; 33; 35
vocabulary, 7; 10; 22; 23; 26; 27; 59; 74
Voice User Interface. *See* VUI
VUI, 3
  design issues, 8–14
  design toolkit, 55
Wickens, 7; 76
wide-band, 21
within-subject variable. *See* variable
Wizard-of-Oz, 26; 53
word spotting, 8; 10; 21
Yankelovich, *et al.*, 13; 76
Zoltan-Ford, 4; 7; 8; 9; 22; 23; 26; 27; 28; 29; 59; 60; 76