A VOICE INTERFACE FOR VTLS

by

Pranav Mehta

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

V.Chachra, Co-chairman
M.Nadler, Co-chairman

C.Nunnally

May, 1989

Blacksburg, Virginia
A VOICE INTERFACE FOR VTLS

by

Pranav H. Mehta

Committee Co-chairmen: Vinod Chachra, Morton Nadler

 Electrical Engineering

(ABSTRACT)

The objective of this study was to develop a voice interface for the on-line catalog of VTLS. Three major components of the system, namely, voice recognition system, text-to-speech synthesizer, and screen review program, were identified. These components were selected after a comparative study of several commercially available systems. Once the components were selected they were integrated to form a complete voice recognition and synthesis system. Using this system, a voice interface was realized to suit the operations of VTLS. A telephone interface for the system was investigated and recommendations were made for future research.
ACKNOWLEDGEMENTS

This work would not have been possible without the vision and inspiration of Dr. Vinod Chachra, who helped and guided me through the various phases of the research while teaching me the importance of dedication and patience in research. I hope that I will always be able to live up to the high standards that he sets.

I would like to thank Dr. Morton Nadler and Dr. Charles Nunnally, who encouraged me and helped me to concentrate on the real issues through their constructive criticism.

VTLS, Inc. funded this project through equipment purchase and by sponsoring me with research assistantship. I am indebted to all of the VTLS staff for their continued support during my association with them.

On a personal note, I would like to thank my parents for giving me the wings to fly while making sure that I was going in the right direction. This work would not have been the same without the love and support of my wife, , who has shown me the beauty of life, love, and honesty.
# TABLE OF CONTENTS

1.0 Introduction ................................................. 1  
   1.1 Introduction .............................................. 1  
   1.2 Objectives of the Study ................................. 3  
   1.3 Organization of the Report ............................. 4  

2.0 System Description ......................................... 6  
   2.1 VTLS - A Library Automation System ................... 6  
   2.2 Command Structure ...................................... 8  
      2.2.1 Global Commands .................................. 8  
      2.2.2 Local Commands .................................. 8  
         2.2.2.1 Paging Commands ......................... 10  
         2.2.2.2 Traverse Commands ......................... 10  
   2.3 VTLS Network of Screens ............................... 10  
   2.4 Screen Formats and Translation ....................... 14  
      2.4.1 Characteristics of VTLS screens .............. 14  
   2.5 Implications of VTLS on the Voice Output  
      Component ............................................. 21  
   2.6 Implications of VTLS on the Voice Input  
      Component ............................................. 23  
   2.7 Outlining the proposed interface ..................... 29  

3.0 The Voice Input Component ................................. 30  
   3.1 The Speech Recognition Unit ............................ 30

Table of Contents...iv
1. INTRODUCTION

1.1 Introduction

A library is a home for the cultural heritage of mankind. In a modern library of today, information retrieval in the shortest possible time frame is vital to its operation. VTLS Library Automation System is one such means for fast and efficient information access. One can utilize it through a terminal in a library with little or no training required. A library database can be accessed also from a remote computer through the use of a terminal or PC with a modem.

A very important patron group for libraries consists of visually impaired and other handicapped people who either cannot read the information on a CRT screen or cannot effectively use a keyboard. The existing keyboard-monitor interface restricts their ability to exploit the potential of the library automation system.

Also, in today's world of automation one desires to accomplish all possible tasks from a single location. A computer network encompassing every home is still a distant future.
On the other hand, public telephone lines have covered even the remotest of areas in the sub-continent. It would be very convenient to be able to access a library database over a telephone. The circulation information pertaining to a book is an integral part of the VTLS data base and retrieval system. It would be very useful to know the availability of a book before starting for the library to get it. With document delivery systems this may not even be necessary. The above proposition gains ground under such conditions.

The issues and thoughts presented above were the major motivations behind the idea of developing a voice interface for the library automation system – VTLS. Commercial Speech Recognition and Speech Synthesis products are readily available. Thus there was no need to develop such systems from scratch. It would clearly be beyond the scope of this work. Therefore, it was decided to use the existing products. A thorough market survey and technical review of the merits and interface capability of the equipment and systems was deemed essential. The primary goal was to develop a workstation which provides an interface to VTLS through voice. The possibility of building a telephone interface was accepted as a secondary target.
1.2 Objectives of the Study

As noted above, the objective of this research and development is to build a voice interface for VTLS on-line catalog. The following steps comprise the proposed plan for development:

1. Study the literature on voice recognition and speech synthesis systems. The goal here is to determine previous work done in these domains.

2. Conduct a comparative study of voice recognition systems available in the market to select one out of them best suited for this project.

3. Conduct a comparative study of speech synthesis systems available in the market to select one out of them best suited for this project.

4. Understand enough of the VTLS system to enable the building of a voice interface for the VTLS on-line catalog.

5. Investigate the performance of the voice recognition system to select an appropriate application vocabulary for the interface.

6. Utilize a Screen Review Program to read the contents of the screen using voice recognition as input and speech synthesis as the output.
7. Integrate all of the above components, namely, voice recognizer, speech synthesizer, application vocabulary, and screen review program, into a working voice interface for VTLS on-line catalog.

1.3 Organization of the Report

Before attempting to tackle the problem, it is important to have a historical background on voice recognition and synthesis. This would provide a better perspective for the work to be done here. A summary of the literature search can be found in the background information furnished in the appendix A.

A detailed comparative study (step 2, section 1.2) of the voice recognition systems available in the market is presented in Appendix B. Six characteristics were identified as desirable in the voice recognition sub-system. These are accurate isolated word recognition, context sensitivity, transparency to the application, flexibility, compatibility with the speech synthesizer and screen review program, and telephone recognition capability. Barring the last characteristic, the voice recognition system by Dragon Systems, Inc. was found to satisfactorily meet these criteria.
Compared to voice recognition, speech synthesis is a mature technology. However, text-to-speech synthesis has yet to come up with human-like speech quality. The main requirement in selecting a speech synthesizer was its compatibility with other system components and a satisfactory output speech quality. The compatibility issues are discussed in chapter 4. A detailed comparison of available products appears in Appendix C (step 3, section 1.2). It was found that for this project, using compatibility, flexibility, output quality, and cost effectiveness as the selection criteria, the synthesizer by AiCom Corp. was an optimum choice.

Chapter 2 gives a brief description of VTLS as a library automation system. This study (step 4, section 1.2) provides vital information in the design of the voice interface.- Chapter 3 describes the investigation of the performance of a speech recognizer and relevant aspects for a voice input component. The tests that were performed are described, results are recorded and analyzed. As a result of these, test vocabulary emerges (step 5, section 1.2). Chapter 4 describes the functions of the speech synthesizer and the screen review program as they relate to the voice interface. All these components are then integrated to form the desired voice interface system. Suggestions for future enhancements are provided in the final chapter.
2. SYSTEM DESCRIPTION

The idea of this thesis was to develop a voice interface for VTLS in order to help visually impaired and other handicapped people access the on-line catalog of VTLS. In the long run such an interface can be used by the general public over commercial telephone lines. This effort requires a study of VTLS as a library automation system. This study would be helpful in making sound decisions on how to translate the keyboard-monitor interface of VTLS into a voice input-voice response interface. The following section describes the very basic aspects about the operation of VTLS that are relevant to this study. The specific problems to be addressed while designing a voice interface are also discussed. A clear direction about what needs to be done emerges at the end of the chapter.

2.1 VTLS – A Library Automation System

The following generic description gives a very basic idea about VTLS system's operation. For detailed information regarding VTLS the instruction manual [1] may be used.
Seven user classes are provided in VTLS. Among them, the class USER has the least capabilities; comprising of 'read only' access to on-line catalog, whereas the MANAGER class has all software capabilities including that of maintaining various passwords. In between, there are other classes with varying degree of access. These classes are reference staff, circulation staff, catalogers etc. The voice interface was developed for the first user class, USER. The operation of VTLS as a system will be covered from the USER viewpoint only.

The main on-line, interactive VTLS program is called LIBBIE. It displays information in two ways:

1. Through screens;
2. Through prompts and messages in response to operator (here, USER) input during specific transaction.

'Screens' are organized displays of bibliographic, holdings and inventory records. Prompts and messages associated with transactions are one or two lines in length and are conversational in nature. The following describes the command structure of VTLS and the relationships among various screens.
2.2 Command Structure

There are many commands in VTLS. Generally four typical key identifiers are used to access record data: Author, Title, Subject and Call Number. All VTLS commands are categorized as either global or local.

2.2.1 Global Commands

These can be entered at any point in LIBBIE and are identified by the presence of a slash (/) character in the first or second position of the command. They are of two types of commands: search commands and processing commands. A USER does not have access to the processing commands. Most frequently used global search commands relate to author, title, subject and call number searches. A short list of basic global search commands is given in Table 2.1

2.2.2 Local Commands

Local commands, can be identified by the absence of a slash. After an activity has been initiated through entry of a global command, local commands appropriate to that activity are available for use. Two types of local commands are available to USER: Paging commands and Traverse commands.
### Table 2.1: VTLS Global Commands

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/(Author name)</td>
<td>Author Search</td>
</tr>
<tr>
<td>T/(Title)</td>
<td>Title Search</td>
</tr>
<tr>
<td>S/(Subject)</td>
<td>Subject Search</td>
</tr>
<tr>
<td>C/(Call-number)</td>
<td>Call-number Search directed to Item screen</td>
</tr>
<tr>
<td>H/(Call-number)</td>
<td>Call-number Search directed to Holdings screen</td>
</tr>
<tr>
<td>M/(Call-number)</td>
<td>Call-number Search directed to MARC screen</td>
</tr>
</tbody>
</table>

A Voice Interface for VTLS...9
2.2.2.1 Paging Commands

As noted before, records stored in VTLS are displayed in 'screens'. Each screen is identified with a heading line that includes the name of the library, software identifier ('VTLS'), and the name of the screen. When the information content of a screen exceeds physical capacity of a standard display monitor, the system partitions a 'screen' into smaller portions called 'pages'. Paging commands essentially facilitate moving forward and backward within multi-paged screens. These commands are valid only if the pages or screens they reference exist for the particular search. A list of paging commands appears in Table 2.2.

2.2.2.2 Traverse Commands

Traverse commands are used to reach other screens related to the current screen. The valid uses of these are explained in the following section. Table 2.3 enlists essential and often used traverse commands.

2.3 VTLS Network of Screens

Data within VTLS is displayed in many different types of
### Table 2.2: VTLS Paging Commands*

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>Displays the last page of the previous screen type</td>
</tr>
<tr>
<td>BROW</td>
<td>Browses through keyword search screens removing all restrictions imposed by search arguments</td>
</tr>
<tr>
<td>NS</td>
<td>Displays the next page of the current screen</td>
</tr>
<tr>
<td>PS n</td>
<td>Displays the nth previous page of the current screen</td>
</tr>
<tr>
<td>SK (nn)</td>
<td>Skips the specified number of entries before displaying the next page of the current screen</td>
</tr>
<tr>
<td>SHOW</td>
<td>Redisplays the current page of the current screen. Any updates are reflected.</td>
</tr>
</tbody>
</table>

* This table gives only a typical set of paging commands from the entire range of powerful commands available in LIBBIE.
Table 2.3: VTLS Traverse Commands

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Displays the Item screen for the uniquely selected work</td>
</tr>
<tr>
<td>CA</td>
<td>Displays the Bibliographic Card Screen for the uniquely selected work</td>
</tr>
<tr>
<td>H</td>
<td>Displays the Holdings screen for the uniquely selected work</td>
</tr>
</tbody>
</table>

* This table contains only a subset of all traverse commands available in LIBBIE.
screens. These screens constitute a network, because it is possible to 'travel' from one screen to other related screens without reentering data.

Initial access to the network of screens is achieved through a global search command. If the system contains any information relevant to the command and search argument(s) specified, VTLS displays an appropriate screen. If no qualifying data are found in the data base, one of the following messages is displayed:

'NO QUALIFYING ENTRIES WERE FOUND'

or

'NO EXACT MATCH -- PLEASE TRY AGAIN BEING LESS SPECIFIC'

If the search succeeds, and there is only one publication related to the selected criterion, VTLS displays the corresponding Item or Item menu screen. On the other hand, if there are more than one publications pertaining to a specific search, an intermediate menu screen lists each publication with the title of the work, the main author, and the publisher, city and date of publication. When the user identifies the desired work, by using one of more of paging commands, and enters the line number corresponding to the publication, VTLS displays the appropriate Item or Item menu
screen. From this screen, the user may access related information about the publication using the local traverse commands listed before.

2.4 Screen Formats and Translation

One major aspect in the design of the proposed voice interface is the capability to translate the VTLS screen content into speech in a controlled manner. Most screen review programs available in the market provide varying functions to access different parts of the screens. The content of the screen is then redirected to text-to-speech synthesizer for voice output.

The following pages contain typical VTLS screen formats that VTLS responds with, in response to a user query [1].

2.4.1 Characteristics of VTLS screens

Several points can be noted after a careful study of these screens as follows:

1. Every screen has a header defining the top of the page.
2. Every header contains the characters 'V T L S'.
3. The right hand corner of the header describes what the
YOUR LIBRARY'S NAME ——— VTLS ——— QUALIFYING AUTHORS

1. 1 Hemingway, Andrew.
2. 1 Hemingway, Ernest, 1899-1961
3. 1 Hemingway, Gregory M.
4. 1 Hemingway, John
5. 1 Hemingway, Leicester.
6. Hemingway, Mary Mon.
7. 2 Hemingway, Mary Walsh.
8. 1 Hemingway, Patricia Drake.
9. 3 Hemingway, Samuel Burdett, 1883-1950
10. 1 Hemingway, Thomas, Keith.

Please enter NEW COMMAND or LINE # of selection

Figure 2.1 VTLS Author Menu Screen
A Voice Interface for VTLS...16

Figure 2.2 VTLS Intermediate Author Menu Screen
Figure 2.3 VTLS Referral Screen
A Voice Interface for VTLS... 18

Figure 2.4 VTLS Item Screen
RELEASE 4.2 DEVELOPMENT - V T L S - BIBLIOGRAPHIC SCREEN

Call Number: 157.8 R29 No. 291337
Title: Use of neutron activating tracers for simulating water and chemical flow through porous media: research project technical completion report / F. G. Heiser ... (et al.). --
Material: xi, 306 p. ; 20 cm. --
Series: (United States. National Technical Information Service. PB; 291337)
Note: Supported by the U.S. Dept. of Interior, Office of Water Research and Technology, under project A-046-PA.
Note: Bibliography: p. 220-229.
Subject: RADIOISOTOPES IN HYDROLOGY.
Subject: RADIOACTIVE TRACERS.
Subject: WATER, UNDERGROUND -- MEASUREMENT.

Enter 'MS' for more

Please enter 'C' for CIRCULATION INFORMATION

Figure 2.5 VTLS Card (Bibliographic) Screen
rest of the screen refers to. For example, in response to an author search initiated by the command 'A/SMITH', if there are multiple records matching the search criterion, the header contains a phrase 'QUALIFYING AUTHORS' which aids the user to keep track of the transaction being processed.

4. Whenever there is a larger number of matching records, VTLS displays the first 16 lines (records, if all are contained within one line). The next line is blank and then in the right hand corner of the next line it displays "'NS' for more", prompting the user to enter Next Screen command to access the next page.

5. In most cases of successful searches, the program prints 'Please enter NEW COMMAND or LINE # of selection' in the line just above the cursor position. The header and this prompt can be considered to constitute the boundaries for the information content VTLS displayed in response to a command.

6. For intermediate screens, although different contents are displayed in a different format under 'Publication for selected criterion' (where criterion = author, title, subject, or call number), the defining boundaries and the logic of 16 lines remain the same.

7. An Item screen contains information specific to a particular copy of a book or serial. It contains the essential details about a book, like, call number,
author(s), title, availability status, hold status, circulation count to-date, etc.

8. If the system is being used from a PC with a standard communications protocol software, VTLS does not necessarily start a new VTLS screen at the top of the monitor screen. Instead, new information is often displayed in response to any command, the screen is scrolled up to accommodate the new data.

2.5 Implications of VTLS on the Voice Output Component

The above observations, coupled with the desired goal of translating useful screen content into speech, lead to the basic requirements for the voice output component of the system. These requirements are defined below:

1. Since the text information to be displayed on the screen varies all the time, a record-and-playback type of speech synthesizer will not work. A good quality text-to-speech synthesizer is required. Capability to use standard set of English grammar rules to construct speech output from a string of ASCII text characters transmitted on its serial input channel is essential.

2. Different users may prefer different speed, inflection, pitch, and other characteristics of the voice generated
by the speech synthesizer. For example, some users can comprehend speech at rate of 800 words per minute. For others, a rate of 250 words per minute is unintelligible. Some may prefer a female voice to a male voice. These preferences require that the voice characteristics be easily adjustable.

3. A screen review program is needed which would allow the user to select a portion of or an item on the screen and send the ASCII value of those character strings to the serial input channel of text-to-speech synthesizer.

4. Since the application program (VTLS) is going to be the front end which interacts with the user, the screen review program has to run in the background. It should be triggered to activate only when desired.

5. As mentioned previously, one complicating feature is the scrolling of screen. This can create a situation where there are two 'VTLS screens' on the 24 lines X 80 characters display but only one of them contains the proper information in regards to the last command. If every new screen were to start at the top, then the screen review program could always start reading from the top left corner of the screen till the end of text without being concerned about whether the information being read is current or not. This scrolling feature requires that the
screen review program be able to differentiate between old and new data and limit itself to the latter.

6. A screen may have 16 items on it. The user may not wish to read them in sequential order. So, skipping lines back and forth should be permitted.

7. The user may be interested in only a particular field for the desired record (e.g. 'availability' of a book). Thus, a direct access to some key data fields is desired.

8. Text-to-speech synthesis for fields like title, author name, subject, etc is comprehensible as mostly they are common english words that can be formulated fairly accurately by the text-to-speech algorithms of the synthesizer. But fields like call number should be spelled out character-by-character as even a phonetic expert would be at a loss while trying to pronounce 'MPD 33 ZTP 411.1'! Thus it should be possible to make the synthesizer always spell certain fields.

2.6 Implications of VTLS on the Voice Input Component

As noted previously, VTLS accepts user commands and responses through a keyboard, parses them, recognizes and then executes them accordingly. Capability to interact with VTLS
using voice input instead of the keyboard would require the following:

A. A system which 'recognizes' human voice commands as spoken into a microphone (for amplification and hence better accuracy).

B. A means to 'understand' what was spoken in the context of the application.

C. And finally, a translator to convert the understood knowledge into information which VTLS can interpret and respond to.

A voice input unit is connected to a speech recognition system. A speech recognition system uses sophisticated signal processing, acoustic parameter extraction, segmentation, stochastic processing and template-matching techniques to recognize an utterance spoken by a user. If VTLS is to be accessible to general public over public telephone network, then this would require that the speech recognition system be able to recognize voice patterns of any human being. Given the current technology, this is not possible. However, enough can be done to still make a useful system. For now, the overall system requirements will be established.
1. The speech recognition system should be speaker independent. If it is speaker dependent, then it should allow for easy switching between users.

2. Whether speaker independent or dependent, at present a recognition system cannot recognize an indefinite number of phrases. Thus, the phrases that it is expected to recognize has to be defined. These phrases will be called the application vocabulary. Different recognition systems have different capacities for the size of the application vocabulary. Thus, an appropriate vocabulary for VTLS application needed to be designed. The selected recognition system should be able to accommodate the desired application vocabulary.

3. A speaker dependent system would require a training session during which a user provides the system with multiple voice templates for the application vocabulary. The system would then store the average of these templates as a word model for each word in the vocabulary. Every time voice input unit is used, the user would have to identify himself. The system then reloads his word models into computer memory and uses them to match against the spoken words for recognition. This is easy enough as long as training session is simple and brief.

4. Once an utterance is recognized, it is important to give it a 'meaning'. VTLS already has such context-sensiti-
vity built into its logic through which it understands and executes or rejects user commands. It will be shown later that such context sensitivity improves the performance of a speech recognition unit. So it would be a good idea to have an application syntax or grammar building capability.

5. It would be attractive to have a system which can recognize continuous speech. But the idea, as much as it is exciting, is also premature and over-ambitious. Firstly, the application considered here can be very well adapted to isolated word recognition. Secondly, and more importantly, the difficulty arises not only in recognizing a continuous sentence but also in imparting semantics to it.

6. One very important aspect is the transparency of the voice interface to VTLS. In other words, it would be undesirable to alter the VTLS software to accommodate the proposed voice interface. Therefore, the new interface should be implemented without disturbing the existing setup. This can be achieved through a mechanism similar to macro commands. When the speech recognizer receives voice input and recognizes it, if it can supply VTLS with appropriate keystroke sequences that a user would have entered from a keyboard to execute the same command. Thus, to VTLS, the new arrangement would be
transparent and it would parse the keystroke sequence in
the same manner and execute the command with no know-
ledge of where these came from.

7. Another feature of practical importance is verbal feed-
back. When one speaks into the microphone attached to a
speech recognition system, it is important to know
whether the system understood the command or mistook it
for something else. In a keyboard-monitor interface the
monitor provides a visual feedback to the user. Since a
two-way voice interface is considered here, this cannot
be employed. Since a text-to-speech synthesizer is al-
ready going to be used for screen translation, it would
be optimum if the speech recognizer also uses the same
synthesizer for verbal feedback to the user. This stra-
tegy would reduce the number of components in the
system, and hence its complexity and cost.

8. Like the screen review program, the speech recognition
program has to run in the background. It will be acti-
vated only when an input is detected on its voice input
channel.

9. The speech recognizer should be compatible with the
screen review program. This is necessary because both of
Figure 2.6 Proposed VTLS Voice Interface
them use the same speech channel to text-to-speech synthesizer and should not conflict with each other. The commands to activate and utilize the screen review program will be included in the vocabulary of the speech recognizer.

10. Finally, keyboard should always be available as an alternative for input purposes.

2.7 Outlining the proposed interface

An outline of the proposed voice interface for VTLS library automation system has emerged from previous discussions. The block diagram in Fig. 2.6 describes the various functions of the interface.
3. THE VOICE INPUT COMPONENT

A performance assessment of the voice input component and the development of the application vocabulary will be described in this chapter. The proposed voice interface consists of three basic components, namely, speech recognizer, text-to-speech synthesizer, and screen reader program. The speech recognition unit represents current speech technology. The reasons for special treatment of its subject matter will become obvious. A test of the effectiveness of the speech recognition unit was conducted. The test set-ups are documented. The test results are tabulated and analyzed. An effective application grammar was developed from the test results. This vocabulary was enhanced (next chapter) to incorporate commands related to the voice response component.

3.1 THE SPEECH RECOGNITION UNIT

The speech recognition unit is a major component of the complete, integrated voice interface for the VTLS system. The performance of the system relies heavily on the recognition accuracy of this unit. It is important that a
determinative measure of the performance of the recognizers be established and used consistently for the evaluation process. Unfortunately as automatic speech recognition is still an emerging technology, there are no standards established to implement tests and determine their usefulness for various purposes. This requires continuing efforts to identify the relative importance of factors which influence performance and to develop and specify definitive test procedures.

In the absence of a common consensus among researchers, vendors, and users, regarding a standard test set-up, the IEEE Working Group on Speech I/O Systems Performance Assessment has outlined several guidelines and caution flags to be considered in designing a test [2]. An effort was made to follow these directions as closely as possible in establishing the test procedure. The remaining sections describe the objectives of the test, the set-up, how various factors influencing performance were accounted for, and discuss the results of the experiment.

3.1.1 The Objectives

Based on the preliminary survey of commercial products available in automatic speech recognition, the choice of
viable products for this particular application was narrowed down. The experiment discussed here was carried out using 'VoiceScribe-1000 Speech Recognition System' by Dragon Systems, Inc. It is a speaker dependent, isolated word speech recognizer. The experiment had two primary objectives:

1. To test the application vocabulary for different users and try to determine the performance of the speech recognizer.
2. To identify, from the results, a confusion matrix between different words in the vocabulary. This information is then used to make the words more distinguishable. The results then give an efficient vocabulary for the application.

3.1.2 The Test Set-up

There are two complementary approaches for designing tests to assess a recognizer's performance. One approach is characterized by the use of a "standard" speech vocabulary and database with no use of syntax to actively control the recognition vocabulary. In this approach, little or no

1. A complete comparative study of available commercial speech recognition systems appears in Appendix B.
effort is made to model the application itself. This approach, while providing valuable comparative performance information, does not directly predict performance in real applications. The second approach involves carefully selecting test conditions in order to simulate a real environment. Syntactic constraint on word sequences is used to enhance the probability of correct recognition. Although this approach may fail to furnish useful information for comparing different applications, it has greater predictive power in analyzing a specific application [2]. It should be obvious that the former approach, while extremely useful in terms of benchmark data and standard results, is too extensive to justify its usefulness for a specific object oriented application as the one considered here.

3.1.2.1 Vocabulary

A vocabulary is a group of spoken words or phrases that a computer recognizes as commands or inputs for an application. The actual performance of any speech recognition system depends upon the subset of the vocabulary that must be distinguished at any stage in the application. Both the number of phrases to be distinguished and the acoustic similarity or complexity of these phrases are critical factors. The VTLS library automation system is utilized by two types
of users; one, the library staff, for their internal functions, like circulation, serials control, cataloging, acquisitions, etc., and the general public, for accessing the online catalog. The vocabulary for the experiment was designed with the latter class of users in mind. A preliminary survey of VTLS commands available to a user was made and as result, a vocabulary of 58 words or phrases was selected for the test set-up. In addition, the vocabulary contained 26 alphabetic letters (to spell out names of authors, titles, subjects, etc.), and 10 digits (to specify call numbers, select a choice from several menu items, etc.). A complete list of the phrases used in the vocabulary appears in Table 3.1.

Monosyllabic words (e.g., the natural alphabet except 'w', no, do, etc.) are more difficult to recognize than longer polysyllabic words. The more complex an utterance the more acoustic information and redundancy it contains. The recognition system is able to use the additional information to distinguish the input from other words in the vocabulary and do it with a higher level of confidence. The phonetic alphabet ('alpha' for 'a', 'charlie' for 'c', etc.) was developed to enhance human recognition in radio transmissions. Pilots and military servicemen use it effectively. Dealing with an automatic speech recognition system is no different. In the

A Voice Interface for VTLS...34
Table 3.1 Vocabulary Used for the Experiment

<table>
<thead>
<tr>
<th>help</th>
<th>g</th>
<th>v</th>
<th>zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>search</td>
<td>h</td>
<td>w</td>
<td>one</td>
</tr>
<tr>
<td>author</td>
<td>i</td>
<td>x</td>
<td>two</td>
</tr>
<tr>
<td>title</td>
<td>j</td>
<td>y</td>
<td>three</td>
</tr>
<tr>
<td>subject</td>
<td>k</td>
<td>z</td>
<td>four</td>
</tr>
<tr>
<td>call-number</td>
<td>l</td>
<td>space</td>
<td>five</td>
</tr>
<tr>
<td>card</td>
<td>m</td>
<td>enter</td>
<td>six</td>
</tr>
<tr>
<td>marc</td>
<td>n</td>
<td>refresh</td>
<td>seven</td>
</tr>
<tr>
<td>holding</td>
<td>o</td>
<td>next-page</td>
<td>eight</td>
</tr>
<tr>
<td>a</td>
<td>p</td>
<td>previous-page</td>
<td>nine</td>
</tr>
<tr>
<td>b</td>
<td>q</td>
<td>previous-screen</td>
<td>point</td>
</tr>
<tr>
<td>c</td>
<td>r</td>
<td>skip</td>
<td>delete</td>
</tr>
<tr>
<td>d</td>
<td>s</td>
<td>browse</td>
<td>cancel</td>
</tr>
<tr>
<td>e</td>
<td>t</td>
<td>circulation</td>
<td>quit</td>
</tr>
<tr>
<td>f</td>
<td>u</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
absence of any semantic knowledge of the language, it is very important for the system to have enough acoustic information to be able to distinguish a word. Pilots use the phonetic alphabet to remove the ambiguity and enhance the efficiency of the conversation. The same vocabulary could be used for the automatic speech recognition applications. However, the normal tendency is to avoid learning additional things particularly if it is complicating. Therefore, although the use of phonetic alphabet seemed the best option, another optimum approach was tried which would incorporate simplicity of natural alphabet and at the same time attempt to increase the speech content in alphabetic letters. This approach required speaking each letter twice in succession (e.g., 'aa' for 'a', 'bb' for 'b', etc). Little training is required to use this method and it also tries to improve the level of acoustic information. Whether this level of acoustic content is enough to achieve desirable recognition accuracy was one of the objectives of the experiment.

3.1.2.2 Grammar

A successful approach to achieving higher performance in recognition is to apply syntactic constraints on the allowable input phrases. These syntactic constraints define
an artificial grammar for the application. It allows recognition of a word only if it occurs in a particular context. The imposition of the syntactic structure partitions the total vocabulary into smaller subsets which are activated at only certain points in the recognition process. The result is a more accurate recognition plus faster response time (there being fewer words to match against at each point). In addition to reducing the size of the active vocabulary, the syntactic structure of a language can also improve the recognition accuracy by separating confusing words into different partitions or subsets.

To design an effective grammar the command structure of the VTLS software was studied carefully (Fig. 3.1). It is apparent that the VTLS application has its own syntax. It would simplify matters if the syntactic structure of the speech recognizer's vocabulary mimics the application syntax. Of course, the constraints on speech input sequences need not be very stringent because an invalid input is detected by VTLS and it gives an error message.

The grammar for the application vocabulary was defined using the VOCL compiler developed by Dragon Systems, Inc. [3]. It
Figure 3.1 Command Structure for VTLS
accepts a regular language as its input. A regular language is generated by a regular grammar which does not permit any explicit or implicit recursion in symbol definitions. For more information on language definition using VOCL compiler the instruction manual [3] may be consulted. The compiler creates a language definition file (LDF) from the regular language (LAN) file. One can also derive a Finite State Machine (FSM) diagram describing inter-relationships of various states of the system during the speech recognition process. The language file used for the experiment is given in Fig. 3.2.

Before leaving the discussion on the test vocabulary, it is worth noting the difference between the total vocabulary capacity (typically a function of total memory available to the system) and effective vocabulary size (typically a function of the structure of the imposed artificial language grammar). For artificially constrained tasks, the average number of alternative words that the system has to choose from at any time is called the "perplexity", or the dynamic branching factor for the imposed artificial language. The experiment discussed here had a total vocabulary of 58 words but the imposition of the grammar resulted in a branching factor of 21 words. Thus the effective vocabulary size was reduced to one third of the total vocabulary size.
/* This language file defines the grammar used for the test experiment carried out on VTLS application vocabulary */

/* Here, SENTENCE is the root definition. In the following definition, SENTENCE is defined as a SUB_SENT followed by indefinite number of LOCAL_COMM or SUB_SENT, which may be terminated at the end by EXIT_COMM to 'complete' SENTENCE. At the beginning of the recognition process the recognizer tries to recognize the root definition completely and since there are no more root definitions it again loops back for completion of SENTENCE. */

SENTENCE = SUB_SENT (LOCAL_COMM, SUB_SENT)* EXIT_COMM ;

/* A SUB_SENT is made up of either a 'help' command followed by END_COMMAND (enter or cancel) or a PREAMBLE ('search') followed by a KEYWORD and terminated by an END_COMMAND */

SUB_SENT = (help END_COMMAND), PREAMBLE KEYWORD END_COMMAND ;

PREAMBLE = search ;

END_COMMAND = enter, cancel ;

/* A KEYWORD is either 'author', 'title', 'subject', or 'call-number', followed by NAME (or A_NAME) which basically contains the alphabet (and digits) to specify arguments for these keywords */

KEYWORD = author NAME *, title NAME *, subject NAME *, call-number TYPE A_NAME ;

Figure 3.2 Language File VTLS.LAN

A Voice Interface for VTLS...40
/* A LOCAL_COMM is one of the commands from the following set (digits, refresh, next_page, etc...) terminated by an END_COMMAND */

LOCAL_COMM = COMM END_COMMAND ;

COMM = DIGIT +, refresh, next_page, previous_page (DIGIT) #,
     previous_screen, skip DIGIT DIGIT, browse, circulation ;

TYPE = (D_TYPE delete) * D_TYPE ;

D_TYPE = card, marc, holding ;

A_NAME = ((NAME *) (point·#) (DIGIT *)) * ;

NAME = a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s,
      t, u, v, w, x, y, z, space, delete ;

DIGIT = zero, one, two, three, four, five, six, seven, eight,
      nine ;

Figure 3.2 Language File VTLS.LAN (cont'd)
3.1.2.3 Test Speaker Population

The manner in which different people speak varies and so does the degree of consistency with which an individual speaker repeats words. Obviously, this has an influence on the performance of the speech recognizer. Variations in speech characteristics of different people reflect differences in age, sex, dialect history, and speech idiosyncrasies; whereas variations in an individual's speech reflect stress, motivation, fatigue, etc. An effort was made to account for these factors. Table 3.2 summarizes different aspects of the test population. For individual speech variations, all speakers were told to make an attempt to simulate the above mentioned conditions during the training session so that when the recognizer averages out the samples, each variation has some representation in the final word model.

It was not possible to select speakers for the tests that are a true representative of the ultimate users of the technology. The VTLS staff does have a good variety of speech and test speaker population essentially consisted of VTLS staff. The user training included a sheet explaining the objectives of the experiment. Also, prior to the test, each

A Voice Interface for VTLS...42
Table 3.2 Test-speakers related data

<table>
<thead>
<tr>
<th>Age</th>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 20</td>
<td>20-29</td>
</tr>
<tr>
<td></td>
<td>30-39</td>
</tr>
<tr>
<td></td>
<td>Above 40</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
</tr>
</tbody>
</table>

Geographical Distribution (%)

<table>
<thead>
<tr>
<th>North-east</th>
<th>South-east</th>
<th>Mid-west</th>
<th>South</th>
<th>West</th>
<th>Foreign</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.23</td>
<td>45.00</td>
<td>14.23</td>
<td>4.6</td>
<td>2.69</td>
<td>19.23</td>
</tr>
</tbody>
</table>

A Voice Interface for VTLS...43
subject was given a brief introduction about the recognizer and its operation.

3.1.2.4 Environmental Factors

The performance of a speech recognizer depends on the quality of the speech signal available via the input channel. The input speech signal is affected by several factors including background noise and transmission channel characteristics. To derive meaningful test results the test-environment was controlled.

The experiment was conducted in a closed, 10' X 8' room to ensure minimum background noise. When microphones are located some distance from the speaker's mouth, reflections from nearby surfaces such as desk-tops, equipment and room walls constitute a form of multi-path interference that can be comparable to increased ambient noise effecting a degradation in the system performance. To avoid this, a unidirectional microphone with headset was used to reduce the distance. Subjects were instructed to keep the microphone about half an inch from their mouth. The test set-up displayed the signal strength on the monitor and subjects were advised to speak loud enough to maintain the amplitude level of the speech signal above 30 dB. Typical values were between 37 dB
- 40 dB. The microphone used, although sensitive, did not have any special features like noise-cancelling or push-to-talk. The set-up also allowed the subject to pause during the training and testing. This preempted the undesired and confusing noises like coughs, stammers, "ers", "ums", etc. made by subjects.

3.1.2.5 Training

Training has two meanings in the context of speech recognition systems. One refers to the process to 'train' the recognizer with a user's speech for a specific application vocabulary. To be unambiguous, this can be termed as 'enrollment'. The second meaning of training refers to the process in which a user becomes familiar with the device or system. Generally, familiarization with device leads to improved performance, and the user learns to adapt to explicit, as well as implicit constraints on the form of the input speech [2]. Both types of training was employed.

The enrollment process was carried out by using the Dragon-LAB utility and the VoiceScribe-1000. To begin the enrollment process, type the following at the DOS prompt:
This will load and initialize the speech driver which remains permanently resident in computer memory until computer is shut off or rebooted. Next type,

VS> LAB <CR>

The software now looks for a file named LABINIT.LAB and executes the commands contained therein. For the current test set-up, this file was modified to suit the application. Figure 3.3 shows the .LAB file used. Additional details about this can be found in the instruction manual [3].

In the first step (enrollment), the user is prompted to speak each word in the test vocabulary five times. All subjects were instructed to speak each word, if possible, in five different ways, e.g., loudly, in a tired tone, flatly, as if questioning. The purpose here is to insure that when an average is taken from these samples to build the voice template for the word, these different tones have their representation in it.

2. Throughout the discussion an underline indicates the user input.
; this is the file to execute the sequence of commands in the
; first experiment designed for Dragon Recognizer

; load the language defined in VTLS3.lfd
language vtls3

; create a new vocabulary file for the user
(macro _create-new ("File name":"User.voc" label)
          (close vocabulary)
          (delete-models)
          (create-vocabulary %1 %2)
) _create-new

; collect tokens
collect ,5

; copy these tokens to a common file for all and then build the
; models for the current vocabulary file and save them
(macro _copy-to-all ()
          (copy-tokens ,,,,all)
          (build)
          (save-models)
) _copy-to-all
score ,,,,,scorel,,,,,,yes

; write the statistics of the scoring to a user specified file
(macro _stat ("File name":"User.sta"
          (statistics %1)
) _stat

Figure 3.3 LABINIT.LAB file used for training
(macro _update-all()
   (open-vocabulary all)
   (adapt)
   (save-models)
   (close-vocabulary)
)
_update-all

; this is the end of the LABINIT.LAB File used

**Figure 3.3 LABINIT.LAB file (cont'd)**
3.1.2.6 Scoring the vocabulary

The scoring of the vocabulary is the process of verifying the recognition capability of the speech recognizer relative to the application vocabulary. There are two approaches to score the performance data for isolated word sequences:

1. Scoring without syntactic constraints.
2. Scoring with syntactic constraints.

The first approach determines the recognizer's ability to recognize words in the application vocabulary without any imposition of application specific grammar. It is an indicator of how distinctly the recognizer is able to differentiate between different words. This is very useful in deriving an efficient vocabulary. The second method is an indicator of the efficiency of the application grammar. It is a true measure of how the speech recognizer will behave in the actual application environment.

The first type of scoring was performed by selecting an option from the main menu of DragonLAB. The system prompts the user to speak a particular word from the vocabulary. After accepting the input from the user, the speech
algorithms works on it as if it did not know what was spoken. Finally, the output of this recognition process is compared against the word it prompted and a score is made if both matched, otherwise a substitution error condition is flagged, i.e., a word was mistaken for something else. The qualitative results of this test are shown in the form of an error matrix in Fig. 3.4. The entries in different rows are the spoken words and the columns represent the recognized words. A '2' in 4th row and 7th column indicates that there were two occurrences (out of 25 samples for as many subjects) where spoken word was 'B' and it was falsely recognized as an 'E'. It is obvious that the error matrix is densely populated in the region of alphabetic words and is quite scattered elsewhere. The results in quantitative terms are tabulated in Table 3.3. It can be noted that without the obvious confusion in recognizing the alphabet, the recognizer has an excellent accuracy of 98.13%.

The usefulness of this approach will be apparent in the following discussion. It can be observed from the error matrix in Fig. 3.4 that two words 'marc' and 'card', which are at the same level in the command structure (Fig. 3.1), are involved in a confusion. This suggests a possible weak point in the performance which can be avoided merely by

A Voice Interface for VTLS...50
| Figure 3.4 Error Matrix for Constraint-free scoring |
Table 3.3 Results of Constraint-free Scoring

Speaker Population = 25
Number of Female speakers = 14
Number of Male speakers = 11

Number of words in Vocabulary = 58
Total words spoken during the test = 58 \times 25 = 1450

Total Number of Substitution Errors = 131
Number of correctly recognized words = 1450 - 131 = 1319

Recognition Accuracy = \frac{1319 \times 100}{1450} = 90.97\%
(all words)

Number of errors involving alphabetic letters = 116

Recognition Accuracy = \frac{785 \times 100}{800} = 98.13\%
(without alphabet)
choosing a different word for the same operation. For example, one can specify to look at a 'item' screen instead of a 'card' screen both of which have the same meaning to VTLS. Similarly, a phrase like 'delete' can be changed to 'backspace' which in DOS environment means the same, i.e., deleting the last character entered. Such instances will be taken into account in the final design of the application vocabulary.

The second type of scoring was performed with the grammatical constraints relative to the VTLS application. As before, the option is selected from the DragonLAB menu. For this test, the users were given a predetermined, simulated command sequence during a typical VTLS session. Figure 3.5 shows the sequence used. Recall that during the enrollment the alphabet was spoken as 'aa', 'bb', 'cc',..., etc., so here for the recognition process, the letters of the alphabet were spoken similarly. The error matrix in Fig. 3.6 qualitatively depicts the results while quantitative yields are displayed in Table 3.4. In comparing the results in Fig. 3.6 with that of Fig. 3.4 one can see that the number of words involved in a confusion is reduced (from 46 to 43), although marginally, when syntactical constraint was in effect. Also, the number of letters of alphabet involved in confusion rose (from 22 to 25). This can be attributed to
1. help enter
2. search author D U K A K I S space M I K E enter
3. search subject P O L I T I C A L space F R E N Z Y cancel
4. search title R E P U B L I C enter three enter circulation enter previous-page enter next-page enter refresh cancel skip two six enter next-page enter refresh cancel previous-screen enter browse enter help enter search author P O I N T delete D E X T E R space J O H N enter
5. search call-number card G Q zero zero seven point four five eight nine enter
6. search call-number holding delete marc W W two point V one enter

**Figure 3.5 Test Command Sequence**
|   | E | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 1 | -E-E |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2 | ENTER |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3 | A |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4 | B |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5 | C |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6 | D |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7 | E |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8 | F |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9 | G |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10 | H |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11 | I |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12 | J |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 13 | K |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 14 | L |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 15 | M |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16 | N |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 17 | O |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 18 | P |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 19 | Q |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20 | R |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 21 | S |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 22 | T |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 23 | U |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 24 | V |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 25 | W |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 26 | X |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 27 | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 28 | Z |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Figure 3.6 Error Matrix for Constrained scoring
Table 3.4 Results of Constrained Scoring

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker Population</td>
<td>25</td>
</tr>
<tr>
<td>Number of Female speakers</td>
<td>14</td>
</tr>
<tr>
<td>Number of Male speakers</td>
<td>11</td>
</tr>
<tr>
<td>Number of words used in the test</td>
<td>115</td>
</tr>
<tr>
<td>Total words spoken during the test</td>
<td>2875</td>
</tr>
<tr>
<td>Total Number of Errors</td>
<td>322</td>
</tr>
<tr>
<td>Number of correctly recognized words</td>
<td>2553</td>
</tr>
<tr>
<td>Recognition Accuracy (all words)</td>
<td>88.8%</td>
</tr>
<tr>
<td>Number of errors involving alphabetic letters</td>
<td>302</td>
</tr>
<tr>
<td>Number of words spoken excluding alphabet</td>
<td>1525</td>
</tr>
<tr>
<td>Number of words correctly recognized (excluding alphabet)</td>
<td>1505</td>
</tr>
<tr>
<td>Recognition Accuracy (excluding alphabet)</td>
<td>98.7%</td>
</tr>
</tbody>
</table>

\[ \text{Recognition Accuracy} = \frac{2553 \times 100}{2875} = 88.8\% \]

\[ \text{Recognition Accuracy} = \frac{1505 \times 100}{1525} = 98.7\% \]
the fact that in actual practice, compared to first method of formal scoring, the subjects tend not to articulate the words as distinctly. Two opposite forces are in action in this situation. The 'realness' of the simulated session makes people more prone to errors while the segmentation of the vocabulary through grammar tends to improve the recognition accuracy. As we see it, the former effect is prevalent in case of alphabet while the latter predominates in the other phrases, yielding a smaller error-confusion matrix. Substitution errors such as 'next-page' mistaken for 'hh', 'subject' mistaken for 'skip', 'quit' mistaken for 'point', etc., which occurred in the first test, were eliminated by partitioning these words in different subsets of the vocabulary through grammar. Table 3.3 indicates that recognition accuracy overall dropped to 88.8% (from 90.97% for the first test). This is because of tremendous increase in substitution errors for the alphabet. On the other hand, recognition accuracy for words other than the alphabet improved, as expected, from 98.13% previously to 98.7%.

3.1.3 Summarizing tests on recognizer

The results of the investigation into the speech recognizer's performance was seen as very encouraging. The results with the initial vocabulary, and the simple grammar design
were very promising. Some changes in the vocabulary would improve the performance even more. One of these changes was to employ the phonetic alphabet instead of speaking a letter twice in succession. Although this improves on monosyllabic letter pronunciation, it does not result in an acceptable degree of accuracy or performance. Table 3.5 shows the phonetic alphabet as used by pilots. One obstacle to this approach is the need to memorize the new alphabet. But a quick pass through the list of words would reveal that they are very familiar and the entire alphabet can be memorized in only a few readings.

3.1.4 Transparency to VTLS

The next key element for the recognition system is to communicate the understood phrase to VTLS in a manner which requires no changes to the VTLS software. This is accomplished by the DragonKEY utility supplied with the VoiceScribe-1000. The DragonKEY achieves the above function by using overlays. An overlay is a file that contains commands understood by DragonKEY [4]. The design of an overlay starts with the language file similar to one in Fig. 3.2. In addition to the grammatical structure for the vocabulary this file contains information regarding the keystroke sequences to be generated and supplied to VTLS upon
Table 3.5 Phonetic Alphabet

<table>
<thead>
<tr>
<th>Letter</th>
<th>Phonetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>alpha</td>
</tr>
<tr>
<td>b</td>
<td>bravo</td>
</tr>
<tr>
<td>c</td>
<td>charlie</td>
</tr>
<tr>
<td>d</td>
<td>delta</td>
</tr>
<tr>
<td>e</td>
<td>echo</td>
</tr>
<tr>
<td>f</td>
<td>foxtrot</td>
</tr>
<tr>
<td>g</td>
<td>golf</td>
</tr>
<tr>
<td>h</td>
<td>hotel</td>
</tr>
<tr>
<td>i</td>
<td>india</td>
</tr>
<tr>
<td>j</td>
<td>juliet</td>
</tr>
<tr>
<td>k</td>
<td>kilo</td>
</tr>
<tr>
<td>l</td>
<td>lima</td>
</tr>
<tr>
<td>m</td>
<td>mike</td>
</tr>
<tr>
<td>n</td>
<td>november</td>
</tr>
<tr>
<td>o</td>
<td>oscar</td>
</tr>
<tr>
<td>p</td>
<td>poppa</td>
</tr>
<tr>
<td>q</td>
<td>quebec</td>
</tr>
<tr>
<td>r</td>
<td>romeo</td>
</tr>
<tr>
<td>s</td>
<td>sierra</td>
</tr>
<tr>
<td>t</td>
<td>tango</td>
</tr>
<tr>
<td>u</td>
<td>uniform</td>
</tr>
<tr>
<td>v</td>
<td>victor</td>
</tr>
<tr>
<td>w</td>
<td>whiskey</td>
</tr>
<tr>
<td>x</td>
<td>x-ray</td>
</tr>
<tr>
<td>y</td>
<td>yanky</td>
</tr>
<tr>
<td>z</td>
<td>zulu</td>
</tr>
</tbody>
</table>
the recognition of a symbol in the vocabulary. Also, it contains special commands called VCOMs (for Voice COMmands) which are like DOS commands and are executed by DragonKEY when it loads an overlay in memory. These aspects will become clear from the following discussion. Figure 3.7 illustrates the language file used to create an overlay called VTLS4. To create the overlay a macro language compiler called VOCL is used as follows:

VS> VOCL VTLS4 <CR>

This creates an overlay named VTLS4.LDF. A few points can be noted regarding the language file:

1. This language file incorporates the changes proposed in the previous section.
2. The modified structure of VTLS commands is shown in Fig. 3.8.
3. The symbol 'scratch-that' in parentheses beside the root symbol SENTENCE signifies that it is always a part of the 'active' vocabulary. It replaces the 'cancel' symbol in the previously defined vocabulary.
4. The portion of the file after the grammar definition effects the creation of desired keystroke sequences. For example, upon recognition of word 'circulation' this
/* This is the language definition file (VTLS4.LAN) for VTLS application. It also generates appropriate keystroke sequences for various commands. */

/* The following is the root definition */

SENTENCE [scratch-that] = (LOCAL_COMM, SUB_SENT)* EXIT_COMM ;

SUB_SENT = (help END_COMMAND), PREAMBLE KEYWORD END_COMMAND ;

PREAMBLE = search ;

END_COMMAND = enter ;

EXIT_COMM = quit ;

KEYWORD = author NAME *, title NAME *, subject NAME *,
            call-number TYPE A_NAME ;

LOCAL_COMM = COMM END_COMMAND ;

COMM = DIGIT +, refresh, next_page, previous_page (DIGIT) #,
      catalog, last_screen, skip DIGIT DIGIT, browse, circulation,
      HOLD_NO_SLASH ;

/* TYPE = (D_TYPE delete) * D_TYPE ; & & commented out */

TYPE = item, marc, HOLD_WITH_SLASH ;

A_NAME = ((NAME *) (point #) (DIGIT *)) * ;

NAME = alpha, bravo, charlie, delta, echo, foxtrot, golf, hotel,
      india, juliet, kilo, lima, mike, november, oscar, poppa, quebec,
      romeo, sierra, tango, uniform, victor, whiskey, x-ray, yankee,
      zulu, space, BACK_SPACE ;

DIGIT = zero, one, two, three, four, five, six, seven, eight, nine ;

Figure 3.7 Language File VTLS4.LAN
BACK_SPACE = backspace;

'HANDLE_WITH_SLASH "H/" = holdings;
HANDLE_NO_SLASH "H" = holdings;

/* define the keystroke sequences */

help "HELP";
search "s-fl";
author "A/";
title "T/";
subject "S/";
call-number "s-fl";
item "C/";
marc "M/";
refresh "SHOW";
next_page "NS";
previous_page "PS";
catalog "CA";
last_screen "BA";
skip "SK";
browse "BROW";
circulation "C";
point ".";
alpha "A";
bravo "B";
charlie "C";
delta "D";
echo "E";
foxtrot "F";
golf "G";
hotel "H";
india "I";
juliet "J";
kilo "K";
lima "L";
mike "M";
november "N";
oscar "O";
poppa "P";

Figure 3.7 Language file VTLS4.LAN (cont'd)
/* defining the Vocms */
#define Vcom  "bind s-f9 microphone on
           bind s-f10 microphone off
           bind s-f4 backup backspace
           bind /echo enter reset
           bind s-f1 beep happy
           bind s-f2 beep sad
           bind s-f3 beep error" ;

/* end of the language file VTLS4.LAN */

Figure 3.7 Language file VTLS4.LAN (cont'd)
Figure 3.8 Modified Syntactic Structure for VTLS
overlay generates and passes the keystroke 'C' to VTLS which is the standard keyboard entry from the item screen to request circulation information of a book. Keypresses like <ENTER> and <BACKSPACE> are defined as 'enter' and 'backspace'. These are the special names DragonKEY uses for such special keypresses.

5. The last section of the file defines the VCOMs mentioned earlier. Here the VCOM BIND is used which associates any vcom with a single keystroke. Thus, pressing <shift><F9> keys together would execute a vcom to turn the microphone ON while pressing <shift><F10> would turn the microphone OFF. If the user wishes to speak with someone other than the recognizer it might be a good idea to turn the microphone off because if it remains ON, the recognizer will try to analyze the input and on failing to recognize it, will keep on giving error messages.

6. It can be seen that when the word symbols 'search' and 'call number' are recognized, the keystroke generated is <shift><F1>, which is then 'bound' to vcom 'beep happy'. This is because these symbols are used in the grammar to effectively limit the active vocabulary at certain points during the recognition process and do not have a logical keystroke sequence to present to VTLS. At the same time, it is important for the user to have some
assurance that they were recognized properly. When a word symbol is recognized with a high level of confidence, the 'beep happy' vcom will sound a happy tone from the keyboard. This feature will be removed from the final design when a speech synthesizer will ensure recognition through verbal feedback.

7. One very important aspect relates to definition of the symbol 'scratch-that'. Recognition of this symbol generates the keystroke <shift><F4> which is then bound to vcom 'backup backspace'. This vcom requests that DragonKEY generate enough "backspace" keystrokes to erase characters generated by the previously spoken and understood command. Not only that, it also "backtracks" one state in the state table of the vocabulary grammar. For example, if symbol 'next-page' is recognized, DragonKEY produces 'NS' to present to VTLS and the grammar is positioned so as to accept symbol 'enter' (of course, 'scratch-that' is always active). Now if 'scratch-that' is recognized as the next symbol, not only does DragonKEY remove the 'NS' characters from the screen and from the keyboard queue to be sent to VTLS, but it also repositions the grammar to accept any of the local commands (including keywords or 'help'). This is a very useful feature unique to DragonKEY.
8. The last problem is of synchronization between the DragonKEY and the application program. After the recognition of a local command, like 'next-page' or 'refresh', or at any point, in the syntactical structure, if the recognizer is looking for the word symbol 'enter' to be spoken, and the user by any chance hits the <ENTER> key on the keyboard, then VTLS will get the carriage return that it is awaiting and will execute the command and expect a new command while DragonKEY is still looking for the command 'enter'. In this case, it is required to communicate to DragonKEY that an <ENTER> was pressed from the keyboard. Also, in most cases <ENTER> terminates a command. So it would be a good idea to reset the grammar in this event. This is achieved by "vcom bind /echo enter reset". Thus, whenever <ENTER> is pressed DragonKEY intercepts it, returns to the beginning of the root definition of SENTENCE and also passes the keystroke to the application program through /echo option.

3.1.5 Multiple Users

One of the requirements of different voice input component was the ease of switching between users. For a speaker dependent system such as VoiceScribe-1000, it is expected that
every user who desires to use the voice interface should have his own vocabulary file which may be created using DragonLAB utility. In DragonKEY, to switch from one vocabulary file to another, one must first 'unload' all the overlays from the memory, then open the desired new vocabulary file and then reload the overlay into the computer memory. The following commands (vcoms) included in a batch file called VTLS.BAT perform this task:

VCOM OVERLAY /UNLOAD
VCOM VOCABULARY %1
VCOM OVERLAY VTLS4 /PERMANENT

The /PERMANENT option makes the overlay VTLS4 permanently resident until the computer is rebooted or it is intentionally 'unloaded' for a new user. The batch file may be started as follows:

VS> VTLS 'username' <CR>

At this point, a voice-input and video-output interface is realized and can be used as a "hands free" environment. The next chapter describes how the integrated voice interface is accomplished.
The voice response component of the voice interface will include the text-to-speech synthesizer as well as the screen review program. Detailed descriptions of each will be followed by the actual development of the proposed voice interface. A successful implementation of the voice interface will emerge at the end.

4.1 THE TEXT-TO-SPEECH SYNTHESIZER

Appendix C contains a comparative study of some of the commercially available text-to-speech synthesizers. Although 'human-like' voice is still in the future, these synthesizers produce fairly good voice quality. As previously stated, one of the criteria in the selection of a speech synthesizer was its compatibility with the proposed integrated voice interface. This requirement along with its cost effectiveness led to the selection of ACCENT text-to-speech synthesizer designed by AiCOM Corporation, San Jose, California. Its features are listed in Appendix C but some of them will be highlighted here for easy reference.
ACCENT is a plug-in board for the IBM PC with an on-board microprocessor to avoid usage of the CPU on the PC. It has an 800 byte wrap-around text buffer. A standard RS-232C serial port is provided for interface with a computer, terminal, or printer, and the user can select either the PC or the RS-232C as Accent's host. This is important for the desired set-up. Also, it has two speech modes: the text mode and the spell mode. The text mode is the normal operation mode. It converts text into intelligible speech employing grammatical rules, while the spell mode is required to read out call numbers for bibliographic records as discussed above. And through software commands, which are presented as escape sequences, speech parameters such as average pitch, voice characteristics, speech rates, etc. can be controlled. Usually, the incoming text in ASCII format is stored in the text buffer and will not be spoken until a sentence punctuation mark is received. But there is also a programmable automatic time-out feature. If no input is received within this predetermined time interval, speech is initiated. The time-out interval can be adjusted, through software commands, from 0.04 to 20 seconds. This is a useful as the text input may not have any punctuation marks. For more information on the software commands the user's manual should be used [5].
One major obstacle with various speech synthesizers was the conflict in hardware configuration with the speech recognizer. The problem arises out of the limited Interrupt Request lines available in an IBM PC on which the set-up was implemented. Figure 4.1 lists the interrupt line assignments for the PC/XT. Notice that there is only one interrupt line available (IRQ2) for additional hardware. If only one asynchronous communication port is required then possibly one of the IRQ3 or IRQ4 lines may be used. Now, one asynchronous communication port is definitely needed for communication between the PC and the host computer which runs VTLS. One interrupt line is also required for the speech recognizer. Some of the text-to-speech synthesizers available with varied forms of screen reading software option have special speech drivers through which the screen reader redirects the screen information to the text-to-speech converter. These devices, in turn, require an interrupt request line as they communicate with the converter via the I/O channel bus of the PC. This would suffice if verbal feedback of the spoken commands were not needed. But to implement it, the recognizer has to redirect the keystrokes it generated, to the text-to-speech converter. This requires a device driver software specific to the synthesizer. Now, the VoiceScribe-1000 provides a generic speech driver software that sends the generated text to a serial communication port. This port
<table>
<thead>
<tr>
<th>Line</th>
<th>Int. No. (hex)</th>
<th>Vector Address (hex)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>8</td>
<td>20-23</td>
<td>Timer Interrupt</td>
</tr>
<tr>
<td>IRQ1</td>
<td>9</td>
<td>24-27</td>
<td>Keyboard Interrupt</td>
</tr>
<tr>
<td>IRQ2</td>
<td>A</td>
<td>28-2B</td>
<td>Reserved (available)</td>
</tr>
<tr>
<td>IRQ3</td>
<td>B</td>
<td>2C-2F</td>
<td>Asynchronous COMM2 Interrupt</td>
</tr>
<tr>
<td>IRQ4</td>
<td>C</td>
<td>30-33</td>
<td>Asynchronous COMM1 Interrupt</td>
</tr>
<tr>
<td>IRQ5</td>
<td>D</td>
<td>34-37</td>
<td>Disk Interrupt</td>
</tr>
<tr>
<td>IRQ6</td>
<td>E</td>
<td>38-3B</td>
<td>Diskette Interrupt</td>
</tr>
<tr>
<td>IRQ7</td>
<td>F</td>
<td>3C-3F</td>
<td>Printer Interrupt</td>
</tr>
</tbody>
</table>

**Figure 4.1 Interrupt Line Assignments for the PC/XT**
can then be connected to the serial input of most synthesizers and thus converted into speech. But, this would necessitate a second serial communication port and hence another interrupt request line. This creates a situation where there are three interrupt lines needed but only two are available.

This conflict was solved with ACCENT and a suitable screen review program. Although Accent card plugs into the PC, it only draws power from it. The text input it receives is through its serial input port which is connected to an asynchronous communication port. The screen review program described in the next section would redirect the screen content to the same serial port and so would the speech recognizer. No special interrupt line is needed for the synthesizer as it does not use the PC I/O channel bus for its operation.

4.2 THE SCREEN REVIEW PROGRAM

There are many types of sophisticated screen review programs available. Some of these are dedicated to work with specific speech synthesizers while some are intended to be generic. The latter type usually directs the text on the screen to the synthesizer through a serial communication port. Many of these cater to specific design aspects of various software
products. The criteria used to select a screen review program, or a screen reader, as it is sometimes called, were simplicity and cost. Simplicity is interpreted as just enough functionality required by the considered application with no overheads. This naturally results in lower cost. Also, it avoids a potential problem of exceeding memory limits on an IBM PC. Since the proposed voice interface was intended to be a part of a basic, normal PC installation with typically 512K of dynamic memory, this is an important consideration. The PC Disk Operating System (DOS) usually occupies about 80-90K of memory. Another 210-230K will be used by the VoiceScribe-1000 speech recognition's memory resident software for the application vocabulary size. This leaves only 190-210K memory left for all other applications running on the PC.

The screen reader program that seemed to fit the above criteria is called 'FREEDOM1'. It is developed by Interface Systems International of Portland, Oregon. It occupies only 16K of system memory and is reasonably priced. It has some very powerful features. Its basic function is to redirect the ASCII characters in the video screen buffer in the PC memory to the serial port of the ACCENT synthesizer through an asynchronous communication port. It is oblivious to how the characters got to the CRT screen. It is concerned only
with what is actually there at a particular moment. This is the key to its usefulness. On the other hand, this also becomes its limitation. As was noted in a previous discussion, when VTLS displays various screens on a PC terminal the screen scrolls up. If two successive screens do not have enough information to fill the CRT display, there may arise a situation where there are two VTLS screens on the CRT display and Freedoml has no obvious way of knowing which part of the screen is relevant. This limitation, however, is overcome by some of the other powerful features supplied with it. For complete details of these features the owner's manual may be consulted [6].

Freedoml is a memory resident program which once invoked will remain in the computer memory until the computer is rebooted. It is activated from within an application program by pressing the '˜' (tilde) key on the keyboard and is exited by pressing the <Esc> key. Once it is activated, various Freedoml commands can be used to read out the screen in a controlled manner. Some of the commands used are listed in Table 4.1. The following definitions will help in understanding the commands:
### Table 4.1 FREEDOM1 Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Locate host cursor</td>
</tr>
<tr>
<td>U</td>
<td>Move Up one line</td>
</tr>
<tr>
<td>&lt;CR&gt;*</td>
<td>Read one line</td>
</tr>
<tr>
<td>R</td>
<td>Move right one word</td>
</tr>
<tr>
<td>H</td>
<td>Move to home position (active window)</td>
</tr>
<tr>
<td>^L</td>
<td>Move left one column</td>
</tr>
<tr>
<td>D</td>
<td>Move Down one line</td>
</tr>
<tr>
<td>V</td>
<td>Reverse mode toggle</td>
</tr>
<tr>
<td>F(string)&lt;CR&gt;</td>
<td>Find a 'string' of characters</td>
</tr>
<tr>
<td>Mm</td>
<td>Set a marker 'm' at the current review cursor position</td>
</tr>
<tr>
<td>Gm</td>
<td>Go to marker 'm'</td>
</tr>
<tr>
<td>^Wnmm</td>
<td>Create a window number 'n' between two given markers</td>
</tr>
<tr>
<td>^O(string)&lt;End&gt;</td>
<td>Direct Keyboard output</td>
</tr>
</tbody>
</table>

* indicates that this command will initiate speech output.
**Review cursor:** It is an imaginary pointer to the spot on the computer's screen where Freedom1 will begin reading at the next reading command.

**Host cursor:** This is the location on the screen where the next keystroke typed into the application program (here, VTLS) will be displayed.

**Window:** A window is a rectangular area on the screen which can be defined and isolated for review.

For the VTLS application, the area between upper leftmost corner and lower rightmost corner is defined as window #1. Freedom1 supports, through its utility program FREEU, the creation of macro commands to execute multiple commands with one keystroke. This feature was used to perform the desired translation of the VTLS screens. Table 4.2 lists definitions of various macro commands used in the application. The logic behind these macros and their intended use will be explained briefly.

VTLS responds to a user command with a screen and/or a prompt. It was noted earlier that the line just above the
Table 4.2 Definition of FREEDOM Macro Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Key-Stroke Assigned</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>review</td>
<td>&lt;alt-1&gt;</td>
<td>- , 1, C, U, &lt;CR&gt;</td>
</tr>
<tr>
<td>read-next</td>
<td>&lt;alt-3&gt;</td>
<td>&lt;CR&gt;</td>
</tr>
<tr>
<td>read-last</td>
<td>&lt;alt-4&gt;</td>
<td>1, C, U, U, U, M, Y, 2, G, Y, &lt;CR&gt;</td>
</tr>
<tr>
<td>before-last</td>
<td>&lt;alt-5&gt;</td>
<td>U, U, &lt;CR&gt;</td>
</tr>
<tr>
<td>repeat</td>
<td>&lt;alt-6&gt;</td>
<td>U, &lt;CR&gt;</td>
</tr>
<tr>
<td>pause</td>
<td>&lt;alt-7&gt;</td>
<td>'O, &lt;Esc&gt;, =, Q, &lt;End&gt;</td>
</tr>
<tr>
<td>continue</td>
<td>&lt;alt-8&gt;</td>
<td>'O, &lt;Esc&gt;, =, q, &lt;End&gt;</td>
</tr>
<tr>
<td>exit</td>
<td>&lt;alt-9&gt;</td>
<td>1, &lt;Esc&gt;</td>
</tr>
<tr>
<td>author</td>
<td>&lt;alt-A&gt;</td>
<td>2, H, F, A, u, t, h, o, r, &lt;CR&gt;, &lt;CR&gt;</td>
</tr>
<tr>
<td>title</td>
<td>&lt;alt-T&gt;</td>
<td>2, H, F, T, i, t, l, e, &lt;CR&gt;, &lt;CR&gt;</td>
</tr>
<tr>
<td>subject</td>
<td>&lt;alt-S&gt;</td>
<td>2, H, F, S, u, b, j, e, c, t, &lt;CR&gt;, &lt;CR&gt;</td>
</tr>
<tr>
<td>status</td>
<td>&lt;alt-U&gt;</td>
<td>2, H, F, S, t, a, t, u, s, &lt;CR&gt;</td>
</tr>
<tr>
<td>skip</td>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>

A Voice Interface for VTLS...78
host cursor is usually a one line prompt (or an error message) instructing the user to take an appropriate action. It is also a logical starting point for the review process because if it were a prompt like 'Please enter a New Command or Line # of selection', it indirectly suggests that VTLS did display some useful information in response to the last command; on the other hand, if it were an error message, such as 'Invalid command - please enter a New Command' or 'No exact match - please try again', it directs the user to take corrective action. In the former case, the user may wish to retrieve the information displayed using a 'screen review command', while in the latter case, he might simply choose to use a new 'VTLS command'. Assuming the first situation, it is logical that the user would want to read the first line of the screen just displayed and then decide the next course of action.

In view of the above logic and some of the requirements set forth in previous chapter, the macro commands are designed as follows:

1. 'Review' is the first command to be used to activate Freedoml. It also reads the line above the host cursor so that one can decide whether to review the screen further or exit and continue with search commands.

A Voice Interface for VTLS...79
2. 'Read-first' contains another macro command <Alt-F> in its definition. <Alt-F> enters window #1, goes to the host cursor, sets the reverse mode and then searches for the string 'VTLS' on the screen. Even though there are two VTLS screens on the CRT display as previously discussed, searching for the header from the bottom ensures that the one found first is the latest one. Once the header is found, the screen type is announced. Then, between the leftmost corner of the next line and the rightmost corner of the line above the prompt line, a window labeled #2 is defined. This now defines the active area on the CRT display pertaining to last search command.

3. Commands such as 'read-next', 'before-last', 'repeat', 'skip', etc. are simple manipulations of the commands in Table 4.1.

4. The definition of 'pause' and 'continue' commands makes use of the direct keyboard output capability of Freedom1. The characters following ^0 are sent directly to the synthesizer. In this case, they are the escape sequence to immediately pause or resume the speech output.

5. Commands such as 'author', 'title', 'subject', and 'status' search for appropriate strings within the
active window defined and then read the corresponding field.

6. For the 'call-number' command, the system searches for the string 'call number' and then switches to the spell mode, reads the text and then the normal text mode is resumed.

7. After retrieving the desired information from the screen, the screen review program has to be exited by using the 'EXIT' command. This is important because it first enters window 1 and then exits the review program. If the <Esc> key is pressed when the review cursor was within window 2, then next invocation of the review program will result in an 'out of range' message.

For a better understanding of these macro commands and their functions, references [5] and [6] should be consulted together.

4.3 THE FINAL, INTEGRATED VOICE INTERFACE

A comprehensive and clearly defined voice interface for the VTLS library automation system emerges when all three sub-components of the system are integrated. Figure 4.2 illustrates the interfacing among various constituents and across the entire system. It is a cumulation of all different ideas
A Voice Interface for VTLS...82
covered in earlier discussions. The revised vocabulary and the grammatical structure are presented in Fig. 4.3. It can be seen that the vocabulary incorporates commands for screen review purposes and the keystrokes generated in association with these commands are the ones defined in Table 4.2. Also, it was deemed important to allow a user to create different commands through the use of phonetic alphabet. Thus, to initiate an author search one may either say 'search author' or say 'alpha slash', both of which will cause the creation of keystrokes 'A/'.

The following steps describe the procedure for setting up the system for actual use. The procedure is divided into easy, simple steps to avoid confusion and ensure proper flow of the operation. Some assumptions and clarifications will be presented first:

A. The VoiceScribe-1000 board has been assigned to interrupt request line 2 (IRQ2) through proper jumper settings as per manual [3] and all program and supporting files for the recognizer are in a directory called VS.

B. The AUTOEXEC.BAT file in the root directory of the IBM PC/XT hard drive has the following line:
/* This is the language file (VTLS5.LAN) for VTLS application vocabulary. It also generates keystroke sequences for various commands. It also incorporates commands for screen review program. */

/* The following is the root definition */

SENTENCE [scratch-that] = (LOCAL_COMM, SUB_SENT, S_REVIEW) *
XIT_COMM ;

SUB_SENT = (help END_COMMAND), PREAMBLE KEYWORD END_COMMAND ;
PREAMBLE = search ;
END_COMMAND = enter ;
EXIT_COMM = quit ;

KEYWORD = K_AUTHOR NAME *, K_TITLE NAME *, K_SUBJECT NAME *,
K_CALL-NUMBER TYPE A_NAME ;

LOCAL_COMM = COMM END_COMMAND ;

COMM = NAME +, DIGIT +, refresh, next_page, previous_page (DIGIT)
#, catalog, last_screen, skip DIGIT DIGIT, browse, circulation,
HOLD_NO_SLASH ;

S_REVIEW = review, (READ_COMM *) exit ;

READ_COMM = top-line, bottom-line, next-line, repeat, one-up,
pause, continue, R_AUTHOR, R_TITLE, R_SUBJECT, status, skip-one,
R_CALL-NUMBER ;

K_AUTHOR "A/" = author ;
R_AUTHOR "a-a" = author ;
K_TITLE "T/" = title ;
R_TITLE "a-t" = title ;
K_SUBJECT "S/" = subject ;
R_SUBJECT "a-s" = subject ;
K_CALL-NUMBER = call-number ;
R_CALL-NUMBER "a-c" = call-number ;

/* TYPE = (D_TYPE delete) * D_TYPE ; & & commented out */

TYPE = item, marc, HOLD_WITH_SLASH ;

A_NAME = ((NAME *) (point #) (DIGIT *)) * ;

Figure 4.3 Language File VTLS5.LAN

A Voice Interface for VTLS...84
NAME = alpha, bravo, charlie, delta, echo, foxtrot, golf, hotel, india, juliet, kilo, lima, mike, november, oscar, poppa, quebec, romeo, sierra, tango, uniform, victor, whiskey, x-ray, yankee, zulu, space, BACK_SPACE, C_SLASH ;

DIGIT = zero, one, two, three, four, five, six, seven, eight, nine, BACK_SPACE, C_SLASH ;

BACK_SPACE = backspace ;

C_SLASH = slash ;

HOLD_WITH_SLASH "H/" = holdings ;
HOLD_NO_SLASH "H" = holdings ;

/* define the keystroke sequences */

help "HELP" ;
item "C/" ;
marc "M/" ;
refresh "SHOW" ;
next_page "NS" ;
previous_page "PS" ;
catalog "CA" ;
last_screen "BA" ;
skip "SK " ;
browse "BROW" ;
circulation "C" ;
point "." ;
alpha "A" ;
bravo "B" ;
charlie "C" ;
delta "D" ;
echo "E" ;
foxtrot "F" ;
golf "G" ;
hotel "H" ;
india "I" ;
juliet "J" ;
kilo "K" ;
lima "L" ;
mike "M" ;
november "N" ;
oscar "O" ;
poppa "P" ;
quebec "Q" ;
romeo "R" ;
sierra "S" ;

Figure 4.3 Language File VTLS5.LAN (cont'd)
tango       "T" ;
uniform     "U" ;
victor      "V" ;
whiskey     "W" ;
x-ray       "X" ;
yankee      "Y" ;
zelu        "Z" ;
space       " " ;
backspace   "'backspace'" ;
slash       "/" ;
enter       "'enter'" ;
zero        "0" ;
one         "1" ;
two         "2" ;
three       "3" ;
four        "4" ;
five        "5" ;
six         "6" ;
seven       "7" ;
eight       "8" ;
nine        "9" ;
scratch-that  ":s-f4" ;
quit        ":QUIT'enter'" ;
review      "'a-1'" ;
top-line    "'a-2'" ;
bottom-line    "'a-4'" ;
next-line    "'a-3'" ;
repeat       "'a-6'" ;
one-up       "'a-5'" ;
pause        "'a-7'" ;
continue     "'a-8'" ;
exit         "'a-9'" ;
status       "'a-u'" ;
skip-one     "D" ;

/* defining the Vooms */

#define Voom        "bind s-f9 microphone on
                  bind s-f10 microphone off
                  bind s-f4 backup backspace
                  bind /echo enter reset";

/* end of the language file VTLS5.LAN */

Figure 4.3 Language File VTLS5.LAN (cont'd)
DRAGON /I2

This initializes the recognizer speech channel. The command option /I2 indicates that when a voice input is detected, the recognizer will cause an interrupt through IRQ2.

C. A vocabulary file for the user intending to use the interface has already been created with help of DragonLAB.

D. Once the language file is created for the application, it should be converted to an overlay by using the VOCL macro language compiler as follows:

```
VS> VOCL VTLS5 <CR>
```

The compiler will print some dots on the screen and then signal the completion of the compilation and creation of an .LDF file.

E. The program files for the screen review program Freedom1 reside in a directory called FREE.

F. The macro commands defined for screen review are stored in a parameter file called PTEST.

G. The voice input component is capable of recognizing isolated word sequences, hence all commands should
be spoken with about half a second pause between them.

H. The ACCENT speech synthesizer comes with a factory default setting of 1200 baud rate with PC as the host. These have to be changed to 9600 baud rate and a configuration for RS-232C as the host respectively according to the manual [5].

I. Hardware components are to be connected as shown in Fig. 4.2.

After observing the above requirements and making appropriate changes, one can proceed to initialize the voice interface as summarized below:

1. Power up or reboot the computer.
2. The DragonKEY is initialized through a batch file called KEYINIT.BAT. One command line in this batch file is:

   `BGDRV /G /C1 /B9600 /PNONE /S1`

   This loads a speech device driver which would deliver the keystroke sequences, generated in response to a voice command, to the asynchronous commu-
communication port 1 at 9600 baud rate with no parity and 1 stop bit. To invoke DragonKEY, type,

FREE> CD\VS <CR>
VS> KEYINIT <CR>

DragonKEY is a memory resident program which once loaded returns the control to DOS. Now to load the vocabulary file for the intended user as well as the language file or overlay for the application; type,

VS> NEW 'username' <CR>

3. The screen review is now to be initialized to work in conjunction with the text-to-speech synthesizer and loaded into memory through a batch file called ACCENT.BAT supplied by the vendor. Type,

C> CD\FREE <CR>
FREE> ACCENT <CR>

The synthesizer will give an initialization message stating that ACCENT is now ready.
At this point both the voice input and voice response components are initialized and are waiting to be activated.

4. Now, a VTLS session should be initiated on the host computer. The communication is accomplished through a common communication software called 'KERMIT' over the asynchronous port 2. More information about logging into a LIBBIE session can be obtained from VTLS instruction manual [1].

5. Once the application program VTLS is entered, all the systems are in a synchronized state to start the operation. First, a VTLS command can be spoken and then subsequent screen review and/or VTLS commands can be carried out as necessary until the session is ended with the QUIT command.

4.4 CONCLUSIONS

A Voice Interface for VTLS library automation system was developed through integration of various commercially available products after a thorough study of the market and the systems aspect of the interface.
5. FUTURE DEVELOPMENTS

During the various stages of the development of the voice interface for VTLS many unknown aspects to the problem were discovered.

It should be obvious that the performance of the system is dependent on some commercially available products which are at the cutting edge of the speech I/O technology. For example, the development of speaker independent speech recognition technique will tremendously impact the scope of the application considered here. Similarly, a search for a better quality and affordable text-to-speech synthesizer is still continue.

One area where a concentrated effort would definitely yield dramatic results is the telephone interface to the system. It was one of the ambitious goals of the project to provide for an interface that would allow a user to access VTLS over telephone lines. Some speech recognition systems do have phone interfaces but all of them offer basically telephone management function. Recently a vendor has started providing a set of functions, which can be called from other languages such as 'C', that, if integrated with a system like this
one, can achieve voice recognition over public telephones. To integrate these callable functions in one program requires knowledge of both the programming language as well as the speech recognition process. The effort required to do this was beyond the scope of this work.

The idea, from the very beginning, was to develop the interface without altering the existing VTLS software and it was shown that a workable system could be developed within this constraint. However, now that the voice interface development is no longer an experiment, a few changes, as outlined below, in VTLS software to accommodate its specific needs would be very beneficial.

1. A typical situation arises when a user has finished a VTLS command and the system is searching the data base for appropriate response. Sometimes, it may take about 3-5 seconds to complete a search assuming a very large data base. If during this interval, the user speaks another VTLS command, then the system ignores it as it looks for a user input only after it prompts the user. The user has no way of knowing that it took VTLS some time to finish the first search and that the second command was ignored. Another glitch surfaces when during the search interval for a VTLS command the user issues a
screen review command. The screen reader, which is running in the background, does not concern itself with how the screen buffer was filled. It will start accessing the contents of the screen buffer and will transmit them to the text-to-speech synthesizer. At the same time, LIBBIE will try to write to this buffer a VTLS screen and/or prompt according to previous command. This causes a conflict between the two, resulting in a disfigured screen. Total unpredictability prevails over the operation of the system under these circumstances.

These problems can be remedied simply by adding a feature to VTLS software which would ring a bell whenever VTLS prompts for a user input. This would send a signal to the user that the search is complete and the program is in synchronization with the commands issued. Furthermore, this feature may be made dependent on a flag set at login and thus can be disabled for other terminals, where some other users may find it annoying.

2. The command to search for a string, as provided with the screen reader, is very straightforward. One has to specify the upper or lower case of the characters being searched explicitly. Thus to read a title on the screen one has to specify 'Title' within the search argument.
when defining a macro command. On various VTLS screens, these labels appear in differing cases, e.g., on an Item screen the label reads 'Title' while on an Intermediate screen, it is 'TITLE'. The screen reader will not be able to recognize both as the same and does not possess enough intelligence to determine which character set to look for during a particular search. Thus, it would help if these labels for different fields can be made uniform across the software.

The text-to-speech synthesizer uses an exception table for certain day-to-day phrases and word symbols, e.g., an input string like 'APT.' will be expanded to 'apartment' or 'DEC.' to 'December'. In VTLS screens, the date fields are displayed as '12Jan86'. Now when this character string is passed to the synthesizer, it will read it as 'twelve jan eighty six'. The expansion to 'january' will be carried out only if there were a period ('.') after 'Jan'. Similar problems can be avoided by building a custom dictionary for such exceptions with the help of a utility provided along with ACCENT synthesizer.

Also, a closer study of the macro command capability of FREEDOM1 screen review program can be undertaken to develop
more powerful and useful macro commands for a better control of screen translations.

It is hoped, in the end, that the above and other such ideas stemming from other enthusiastic minds will keep the process of continuing enhancements to the Voice Interface for VTLS alive and help it serve the society better.
REFERENCES


Both, speech synthesis and voice recognition, technologies have steadily grown in stature and maturity over the years, the former probably faster than the latter in the early stages. Describe here is a sampling of efforts by various groups and individuals toward making it more meaningful to industry and society in general.

Williges and Williges [1] note the basic ideology behind a voice input-output system as... 'Voice input and output are being considered for use in human-computer dialogues primarily because of the need to unload the human's visual and motor information channels'. Further, they identify three major components of a system as speech recognition, speech understanding, and speech response. Computer voice recognition is viewed as an extremely complex problem due to idiosyncrasies of the human voice. The technology is still limited in that most systems are speaker dependent. However, current technological advance suggest that these bottlenecks are no longer insurmountable. Speech understanding involves
presence of appropriate syntactical structure and certain degree of artificial intelligence capable of invoking rules to make logical inferences and solve problems contained in the spoken message. Lane and Harris [2] describe four approaches to generating speech including digitized speech, compressed speech, synthesized speech, and developmental work on narrative speech. Two of these approaches, digitized and synthesized speech are use primarily. Digitized speech produces the highest level of realism because words from human speakers are digitized, stored, and then converted back to analog form for playback, but the vocabulary size and structure is somewhat limited. Synthesized speech, on the other hand, is produced by a programmable signal generator representing isolated phoneme sounds. Subsequently, words are formed by combining phonemes, and realism is enhanced by adding inflection.

While evaluating a speech response system, two key elements are intelligibility and quality. Voiers [3] suggested that to achieve an agreement on the standardization of methods to evaluating speech intelligibility and quality, some agreement should be realized on the definition of these and certain other key concepts. According to him, "Intelligibility", however it is measured, ultimately reflects the total amount of information available to, and assimilated by, a
listener concerning the intended content of a voice message. He also defines "context" generally as the totality of factors capable of affecting a listener's a priori uncertainty regarding the contents of a message. Quality is an extremely elusive concept. After examining, the correlations among several subjective criteria, Nakatani and Dukes [4] concluded that ratings of "fidelity", "quality" (as defined by the "harsh-mellow" continuum), "pleasantness", and "understandability" all tapped a common factor, which they termed "acceptability".

In the above context, Pisoni [5] opines that although the field of voice technology has progressed at a substantial rate over the last few years, the same observation does not apply as easily to questions concerning how human observers process (i.e. perceive, encode and interpret) synthetic speech. How do the various commercially available text-to-speech and voice response systems perform? At the present time, he observes, answer to this is unknown. No systematic comparative evaluations have ever been undertaken to assess the performance characteristics of these systems.

In similar efforts to reach a consensus on standardization for performance assessment for speech recognizer, some interesting thoughts are presented. Van Peursem [6] points out

Literature Review...100
that current methods used to evaluate recognition performance do not offer much assistance in the implementation process. Performance must be considered from the user's perspective. A systems approach is required to address the analysis of performance. Even the most robust speech recognition system may fail due to the human speaker, perhaps the most flexible element in the system. The ability for a speech recognition device to be tuned to the user's expectations, as well as the ease in which it may be done, is an important performance issue.

According to Harris [7], performance standards for any system that comprise a human-machine interface must address two questions: (1) How well does the system perform, either in some absolute sense, or relative to other available systems; and (2) How well must the system perform to be useful to a human? While the first question again leads to a standardization for performance assessment, the second one devolve to predicting and measuring how well a set of representative tasks is performed using a candidate system and measuring the extent to which the human's available mental capacities (sometimes called cognitive resources) are consumed in performing the task.
Pallett in his landmark paper [8] identifies and documents factors influencing automated speech recognition performance. Procedures are outlined that are important in designing and implementing performance tests. Speech related, speaker related, task related, environmental, and other factors are identified that affect recognizer performance. Guidelines are defined to document the tests performed. These procedures and directions were followed as closely as possible while setting up tests for the considered application.

Before undertaking a development project like VTLS voice interface, it is important to look for similar attempts made by others. Baker [9] describes a representative sampling of such successful applications, utilizing a variety of leading vendor products. Examples like a worker sorting packages on conveyer belts, operators checking printed circuit boards for specific defects, or inspectors reporting on defects in a textile plant, all using speech recognizer with a headset, in a "hands free" environment, are cited as the major thrust of the technology in industrial applications. In medical science, a radiology report generator is described to create automated reports for routine, limited vocabulary activities. In one instance, inventory audit using speech recognition system is recorded.
Aldefeld, Rabiner, Rosenberg and Wilpon [10] describe an automated directory listing retrieval system based on isolated word recognition. They note the recent attempts at implementing such a system relying on button pushing on the part of the user. Since the Touch-Tone keyboard does not contain a unique key corresponding to each letter in the alphabet, the button pushing system had some drawbacks, e.g., each key would have to be struck one or more times to uniquely specify the letter. They propose a speaker independent speech recognition system using spoken spelled names in place of pushing buttons. Here, a speech recognizer is followed by a post processor directory search algorithm which provides the key to successful data retrieval and features like automatically detecting and correcting simple anomalies in the spelling of the name, including letter substitutions, inversions, deletions, and insertions. If a conflict in the detected name occurs (e.g., 2 or more names with the same of close acoustic scores), the system automatically requests additional information to help resolve the ambiguity. In evaluational tests on an 18,000 name Bell Laboratories directory, the proposed system found the unique correct name 98.3 percent of the trials, on average, even though the acoustic recognizer provided the correct letters only about 70 percent of the time.
Rollins and Wiesen [11] evaluate a speaker independent automatic speech recognition system which performs a telephone switching function. In the application, company employees travelling around the country could call a number which would allow them to place phone calls to any destination number they entered. The application vocabulary is basically limited to digits 0-9, 'yes' and 'no'. The unsuccessful calls were divided into those made by "knowledgeable" users and "non-knowledgeable" users. "Knowledgeable" users were those who knew how to use the system correctly and who had a serious intention of placing a call but were prevented from doing so by difficulties with the system. "Non-knowledgeable" users had calls that were clearly wrong numbers, Touch-Tone input, illegal authorization numbers, etc. The average percent of calls successfully completed for knowledgeable users rose from 83.7% in the 1980 to 91.7% in the 1981 sample.

It can be observed from the ongoing literature review that voice recognition systems have made advances in different quarters of human life, from factories to hospitals, and are proving their effectiveness and worth. Some attempts have been made at limited vocabulary speaker independent speech recognition also. No literature, however, was found on impl-
emeration of a voice interface to access a library database efficiently. In this regard, the work undertaken is hoped to be a significant step to widen the scope of the speech I/O technology. Factors and guidelines discussed before will influence this design set-up.
References


7. Steven D. Harris, "Voice controlled avionics: programs, progress and prognosis (a case for Holistic Engineering)", Proc. Workshop on standardization for speech I/O Technology, Gaithersburg, Maryland (1982).


APPENDIX - B

A COMPARATIVE STUDY OF

VOICE RECOGNITION SYSTEMS
A number of products based on different ideologies are covered. The idea is to make a comparative study of speech recognition systems (or voice response systems which are covered in the next appendix) in the market and have a feel for what direction the technology is heading into and not to make a statement regarding superiority of a particular product over another and hence names are left out from the following comparative analysis. The product finally chosen for the experiment (VoiceScribe-1000 by Dragon Systems, Inc.) was best suited for the desired end under the given circumstances.

In all, there were 12 voice recognition products reviewed for the project. Products ranged from a simple, workable voice recognizer to highly sophisticated, integrated speech recognition systems. When one refers to this evaluation, keeping the following points in mind would help put it in a proper perspective. The idea was to find a system which has:
a. reasonably large (200 words at least) vocabulary
b. good voice recognition accuracy
c. some kind of verbal feedback capability
d. text-to-speech option or compatibility
e. voice recognition capability over public telephone network
f. reasonable cost

B.1 Product A

Features:

The idea here is to house all of the speech recognition circuits within the keyboard enclosure itself and communicate to the host computer through the keyboard port. When a word is recognized, the keyboard recalls a string of keystrokes previously defined by the user and passes them to the host computer exactly as if they were typed from the keyboard. The concept is illustrated in figure B.1. Other features are:

** Provides isolated word, speaker dependent voice recognition.

** Limited amount of context sensitivity incorporated.

** A vocabulary of 160 words.
Figure 9.1 Functional Block diagram for Product A
** Response time of under 210 milliseconds.
** Has its own microprocessor for speech recognition processing.

Advantages:

** Simple, effective concept.
** Eliminates any modification to existing software on the host.
** Reasonable cost.

Disadvantages:

** Speaker dependency.
** Although for the considered application, a vocabulary of about 100 words is sufficient, limited context sensitivity becomes a handicap.
** No additional serial (or parallel) port available for interfacing with a speech synthesizer for voice feedback purposes.

B.2 Product B

Features:
The concept more or less is the same as shown in the figure B.2. Low end simple, limited vocabulary system as well as sophisticated system with 100 words vocabulary and user definable grammar are available. A very good system with continuous speech recognition capability. It has its own microprocessor for speech processing.

**Advantages:**

** Utility programs available to define application grammar for the vocabulary to improve recognition efficiency.**

** Speech response can be added for verbal feedback.**

** Interfaces available to IBM PC family as well as DEC VAX systems.**

**Disadvantages:**

** Higher costs.**

** Speech synthesizer unit interfaces with recognizer and there is no easy way to send text to it from a screen review program to be able to read the contents of the screen.**

**B.3 Product C**
Figure D.2 Functional Block diagram for Product B

Voice Recognition Systems...114
Features (advantages):

** A simple, workable system.
** Economical.
** Speaker dependent, isolated word recognition.
** Speech Synthesis option is also available.

Disadvantages:

** Incompatibility of text-to-speech to work with screen review programs. Basically to be used for verbal feedback of recognized words or from within the application program to read out messages.

B.4 Product D

Features:

** 1000 words, user definable vocabulary which is one of the largest in the market.
** On-board storage - about 1000 speech patterns can be stored in the on-board memory, freeing the host memory.
** Context sensitivity and subsetting of vocabularies.
** Speaker dependent, isolated word voice recognition.
** Transparent mode supports existing application without modifying the application code.

** Communicates to the host computer via an RS-232C serial line.

** Extensive training program available.

** Voice recognition is dependent mainly on patented hardware and hence more expensive.

Advantages:

** One of the best voice recognition systems in the market according to trade magazines.

** Large vocabulary.

** Definable vocabulary grammar to improve recognition accuracy.

Disadvantages:

** Cost not justified for the considered application which utilizes a vocabulary of less than 100 words.

** No information on compatibility with speech synthesizer and screen review programs.

B.5 Product E
Features:

** Speaker independent voice recognition. The only product which claims speaker independency in recognition for a limited vocabulary. The fundamental approach to voice recognition is different from standard 'template matching' methods. The underlying concept is that of phonetic unit identification. With this technique, the basic units of speech which compose an utterance are automatically identified by the recognizer, thereby allowing a given utterance to be recognized from among the set of words which comprise the recognizer's vocabulary. To achieve this, an accurate method of phonetic segmentation is required that is effective in automatically locating phonetic transitions in utterances regardless of inflection, dialect or speaker peculiarities.

Powerful statistical analyzing techniques applied to digitized speech waveforms of a cross-section of speakers determine the optimum value for reference data that will provide highly accurate classification of speech utterances. These optimum settings are obtained with intended user population, application environment and relevant device characteristics in mind.
** Model aimed for PC based systems can support 16 words.
** Telephone interface for recognition over public switching networks available.
** Several custom designed vocabularies available.

Advantages:

** Speaker independent voice recognition.
** Recognition over telephone lines.
** PC version reasonably priced.

Disadvantages:

** Very limited vocabulary size for the considered application.
** Compatibility with speech synthesizer and screen reader programs doubtful.

B.6 Product F

Features:

** Continuous speech voice recognition. It permits users to input data in natural, continuous stream of words without awkward, constrained pauses.
To interpret continuous speech, a grammar has been employed to specify all meaningful, legal utterances that might be spoken. Through a deferred decision-making approach, the system can recognize each grammar as a unit. This is in contrast to most isolated word approaches that use a branching structure of subvocabularies to specify a recognition sequence. To determine successive subvocabularies, systems using such techniques must recognize each word as it is spoken. Moreover, an early mistake can force the system to take a wrong path through the grammar tree. To avoid this, multiple parallel hypotheses are examined by simultaneously comparing input data with the grammar defined set of all legal utterances. A hypothesis is discarded only after obtaining sufficient evidence against its validity. Thus, system defers reporting recognition until a complete legal sequence is spoken. Also, the algorithm's grammatical constraints operate in both word-sequence directions - that is from left to right and from right to left, so that each word provides clues about the words spoken before it and after it. The system knows the syntactic context of the utterance, so it can eliminate many meaningless interpretations of speech and non-speech inputs such as speaker
interjections and background noise, thus improving accuracy.

** Speaker dependent voice recognition.

** Uses an RS-232 serial line to communicate with the host.

** Has a cartridge slot so that multiple users can use the system by simply switching cartridges and loading their own voice patterns.

** Text-to-speech option available.

Advantages:

** Connected speech voice recognition.

** Multiple user, multiple application usage possible.

** Complete grammar definition for the application vocabulary.

** Verbal feedback is also provided.

Disadvantages:

** Even though an attractive feature, continuous speech recognition is not an essential feature to access a library database which easily adapts to isolated word format. While preempting the need of forced pauses between words as in isolated recognition, continuous voice recognition still forces an unnatural grammatical const-

Voice Recognition Systems...120
raint on the user. It is more appropriate for applications like inventory control, shipping/receiving, etc., where each transaction fits one of the limited set of categories and allows effective use of approach used in the system. While exploring a library data base through VTLS, it is very natural to go back and forth in the grammar structure without really "finishing" the sentence and initiating a recognition by the connected speech algorithm.

** Due to above reasons, the high cost of the product is not justified for VTLS application.

B.7 Product G

Features:

** Mini computer style connected-words (continuous speech) recognizer with PC compatibility through software. It segments the speech into phonetic elements and then recognizes words as sets of phonemes.

** 200 words speaker dependent recognition.

** A speaker independent speech recognition option is available with 13 word vocabulary (digits 0-9, 'oh', 'yes', and 'no').

** Interfaces to a host computer via RS-232 serial link.
** Background noise immunity claimed.

** For verbal feedback, previously recorded voice messages can be manipulated and played back.

Advantages:

** Continuous speech voice recognition.

Disadvantages:

** High cost is not justified for VTLS application.

** No information on use of grammatical constraints in recognition to improve accuracy.

** No text-to-speech option available.

** The software for PC compatibility was not rated highly in one trade magazine review.

B.8 Product H

Features:

** An IBM PS/2 based system mainly aimed at automating functions of a telephone operator.

** Provides various call-in features and allows touch-tone inputs in response to different voice menus presented.
** Voice response is created by re-synthesis of previously recorded messages.

** A caller can record a message as an option.

** Can support up to 24 telephone lines with expansion chassis. Standard feature is 4 such lines.

Disadvantages:

** Basically, aimed for banks and other business applications. The 'modus operandi' is irrelevant for the application considered here as although it provides a 'voice automated system', it does not incorporate voice recognition! Input basically consists of touch-tone signals.

** Very high cost.

B.9 Product I

Features:

** Plug-in boards for IBM PC and compatibles to achieve speech recognition as well as speech synthesis.

** Can pool up to nine sets of user voice patterns for a sense of speaker independence.

** Continuous speech, speaker dependent voice recognition.
** Here also the concept is essentially the same as other systems. Phrases recognized by the recognition algorithm are to the application program as pre-determined keystroke sequences.

** Capacity of 50 words of in the active vocabulary.

** Text-to-speech voice output available. The quality of output, however, is not very good.

** An option available for telephone interface. It is essentially for phone management functions like auto answering, auto dialing, etc. and not for voice recognition over public telephone network.

** Comes with a comprehensive application toolkit for easy development of customized application programs. It contains various run time routines to accomplish functions like speech I/O, speech control, speech synthesis and analysis, text-to-speech, speech recognition, and telephone functions and controls. These routines are accessible from other languages like C, PASCAL, and BASIC.

** Application grammar can be defined to achieve context sensitivity.

** A speaker independent voice recognition option is also available with capability of recognizing digits 0-9 and 'yes', 'no'.

** One of the better systems in the market.

Voice Recognition Systems...124
Advantages:

** A comprehensive system with many capabilities.
** Reasonable cost.

Disadvantages:

** Voice output quality is not acceptable.
** Time required to develop a VTLS voice interface type of application considerable.
** Compatibility with screen review program unknown.

B.10 Product J

Features:

** Speaker dependent, discrete-utterance recognition.
** Voice activated keyboard utility allows one to activate keystrokes by voice.
** Uses a macro-language to define application grammar for the vocabulary set thus achieving context sensitivity.
** Text-to-speech option available.
** Voice record/playback feature provided to furnish standard messages like log-in, log-out messages, error messages, etc.

** Telephone management functions available but no voice recognition over telephones.

** Comes with an application toolkit that provides information needed for experienced assembly language programmers to write application programs for voice record/playback, voice recognition, text-to-speech, telephone management, and communication functions.

Advantages:

** Again, a comprehensive system with many capabilities.

** Reasonable cost.

** A very reputed name in the market.

Disadvantages:

** Voice recognition and text-to-speech options are basically licensed by other vendors. They are not OEM products.

** Considerable time required to develop desired application.
B.11 Product K

This product was obtained from the vendor for testing purposes and thoroughly investigated for performance. It is a very sophisticated voice recognition system with most of the desired features listed above. The highlights and some of its drawbacks will be discussed here.

The system comprises of a voice card for IBM PC and supporting software to facilitate adding a voice interface to an existing application. Its hardware also has an optional telephone interface card which fits on the main voice card to form a single unit which can be installed in one of the long slots in an IBM PC/XT or AT. The system claimed voice recognition over telephone lines through this additional card and supplemental software.

The supporting software allows interfacing between the system and the application program. It permits one to define voice commands, train the recognizer, define specific keystroke sequences which are generated and supplied to an application upon recognition of a specific command, and then use those commands to run nearly any application program using speech. Each voice command, consists of a 'label' and a 'key definition'. Labels are the words the user speaks to cause
key definitions to be sent to the application upon proper recognition. A key definition contains the exact keystrokes to be invoked. Defining the voice commands and training the system for various labels are facilitated by a comprehensive menu driven program. The system also provides eight task words which when recognized perform certain control and verification functions. Voice commands are grouped together in a set. Each set contain a maximum of 64 voice commands. In the case of an application requiring larger vocabulary, a technique called set switching is employed. Set switching also allows combining the sets. Although a powerful approach, inducting a degree of context sensitivity in recognition process, the supported set switching concept has few limitations as follows:

1. Inability to "backtrack" during recognition. For example, say, when system recognizes word 'author' (out of a possible set of words 'title', 'author', 'subject', and 'call number') the set switching designates the set containing alpha-numeric characters as the active set for next recognition. Now if one were to change one's mind and decide to backtrack and go back to initial four options, one cannot simply give a command which would position the 'recognition pointer' one step back in the reco-
ognition tree. This can be accomplished only by making a direct call to make a specific set active.

2. Each set is defined separately and although key definitions for certain voice commands may indicate switching to a particular set during recognition, one does not get a clear idea about the interrelations between different commands and thus an understanding of the application grammar as whole.

As mentioned before, however, the major reason of inquiry into this product was phone interfacing capabilities. The telephone interfacing software, along with its hardware, manages most of the activities related to telephone, e.g., answering, auto dialing, keeping customized directories, etc. Once trained, all these functions can be performed by voice. So it is clear that the system does offer recognition over phone lines. But, the phone software, unlike the recognition software, does not go beyond these basic phone management functions and provide generic training and recognition capabilities for it to be integrated with an application. However, the system comes with a voice library which contains 'C' callable voice functions capable of accomplishing voice training, message recording and playback, recognition, DTMF detections, telephone answering, etc. It requires
considerable programming effort in 'C' to put together these functions to form a logic similar to generic recognition software supplied with the system. To be able to integrate it with an application, like VTLS, necessitates that it should be memory resident, increasing the complexity even further. This program development in itself may be viewed as a major effort and was definitely beyond the scope for the work undertaken. Nonetheless, it is noted that existence of a working, useful phone interface is no longer a distant future.

B.12 VoiceScribe-1000 by Dragon Systems, Inc.

Features:

** A speaker dependent, isolated word speech recognition system.
** Recognition is based on patented software and hence reasonable price.
** Most important part of the software is its macro language compiler that allows one to construct a voice recognition grammar for the application to achieve context sensitivity.
** Allows for "backtracking", as discussed previously, through the syntactic structure by one command.
** Handles expanded memory on PCs to provide 1000 word vocabulary, which is one of the largest in the market.

** A memory resident keyboard driver is supplied that allows one to use voice commands and keyboard simultaneously.

** The system includes prefabricated libraries for many popular software applications.

** It includes pop-up menus that display the currently available commands even from within other applications. In addition, all training, testing, and vocabulary functions are furnished through these menus.

** Supports a mode called adaptive training that facilitates adjusting the system to the inevitable changes in a voice that occur over the passing of months or years.

** The documentation is very detailed.

Advantages:

** A complete, versatile system with large vocabulary and many other attractive features.

** Context sensitivity is available. The macro language compiler to construct application grammar is one of the most sophisticated tools in the market.

** Reasonably priced.

** Excellent documentation support.
** Configured to work with text-to-speech synthesizers and screen reader programs in different ways.
APPENDIX - C

A COMPARATIVE STUDY OF

VOICE RESPONSE SYSTEMS
A COMPARATIVE STUDY OF VOICE RESPONSE SYSTEMS

Although speech synthesis is considered a matured technology, compared to speech recognition, this view is based mainly on the performance of record-manipulate-and-playback type of synthesizers that produce seemingly varied speech derived from a clever manipulation of carefully chosen, previously recorded messages. The output voice quality of text-to-speech synthesis, in which only the text to be spoken is delivered to the synthesizer as input and it has to reconstruct an intelligible speech using knowledge of grammar and phonetics, however, leaves much to be desired. Although intelligibility is lower than one would wish, some vendors are coming forth with better than previous voice quality text-to-speech synthesizers with faster speed, improved response time and many other attractive features. A handful of these are covered in comparative analysis here.

C.1 Product A

Features:
** An external unit with on-board microprocessor and text-to-speech firmware.

** It has a built-in speaker and external volume control.

** Features a 1700+ character buffer to store the incoming characters before initiating the speech.

** Simple commands to control the rate of speech, mode of speech (word by word or letter by letter), pitch and volume levels.

Advantages:

** Simple design and reasonable cost.

Disadvantages:

** Does not have many of the functionalities desired like customized exception dictionary, software control over voice characteristics etc.

** Not much technical information available to determine compatibility with a speech recognizer or screen review program.

C.2 Product B
Features:

** A sophisticated text-to-speech synthesizer with various options.

** The synthesizer acts like a printer or serial device. A memory resident module redirects text being sent to a port to the text-to-speech translator. Speech output is controlled like a sequential output device or file.

** A utility allows the user to create fixed vocabulary files that can be retrieved later on by a number from an application program.

** Features improved inflection patterns and a user defined dictionary.

** It comes with a screen review program that features windows to read only a definite portion of the screen.

Advantages:

** Advanced functions like inflection control, custom designed dictionaries, etc. are available.

** The screen review program is also powerful with attractive features like key macros, fast stop and start speech response, less memory usage, etc.

Disadvantages:
** The output voice quality is not acceptable.

** Very limited technical information available to verify compatibility with a speech recognizer.

C.3 Product C

Features:

** External unit with on-board microprocessor and text-to-speech algorithms based on it.

** 750 character buffer size.

** Uses customized speech synthesizer CMOS chip.

** Acts like a heavily buffered RS-232 serial interface printer as far as the host computer is concerned.

** Selectable baud rate (75 to 9600)

** Easy to use.

** Software controlled speech rate, pitch, etc.

Advantages:

** Simple, easy to use and cost effective.

** One of the veterans in the field and hence a stabilized product.
Pronunciations can be controlled from a program through the use of phoneme tables provided.

Known compatibility with screen review program Freedom1.

Disadvantages:

- The output voice quality is not as good as expected in comparison with other products.
- Limited documentation.

C.4 Product D

Features:

- A plug-in card for IBM PCs which is compatible with PC bus interfaces.
- Real time conversion of text to speech.
- Provides external volume control.
- Features external computer communication via RS-232 serial port.
- Supplied with I/O drivers which are callable from programming languages like C, PASCAL and assembler language.

Advantages:
** One of the better systems in the market.
** Good output speech quality.

Limitations:

Based on its reputation, this product was brought in for in-house test purposes and various configurations were tried with it in conjunction with the speech recognizer and screen review programs. The unit accepts the text to be spoken through PC I/O channel and for that a specific device driver has to be written which then drives the text to the text-to-speech port of the synthesizer. Now, the speech recognizer (VoiceScribe-1000) does have a device driver for this particular product so that any command recognized by the recognizer will be directed to synthesizer for voice feedback. However, the intended use of the synthesizer is not only voice feedback to acknowledge proper recognition, but also reading the information on the screen through a screen reader. Most generic screen readers direct the text on the screen to a serial port which is usually connected to a synthesizer. In case of this product, it has a serial port but it is for communicating with a host computer. During the communication the synthesizer is configured to act as a Data Transmitting Equipment (DTE) while for the screen
reader to be able to function with it, it should be configured as data receiving device (DCE).

C.5 ACCENT speech synthesizer

Features:

** On-board microprocessor.
** 800 byte wrap-around text buffer.
** IBM PC bus compatible; I/O addresses switch selectable.
** Standard RS-232C serial port; baud rate, parity and data bits switch selectable; DTE or DCE set-up jumper selectable.
** Programmable automatic time-out for text input without sentence punctuation from the host; 0.04 to 20 seconds.
** Text mode or spell mode speech.
** Various selections of voice characteristics for different personalities.
** Phoneme input capability can be mixed with text.
** Escape sequences for software commands.
** User definable dictionary.

Advantages:
** A sophisticated, well conceived text-to-speech synthesizer.

** Possesses almost all desired features like configurable host communications, software controlled voice characteristics, phoneme capability, customized exception dictionary, etc.

** This unit was brought in for testing purposes and integrated with selected speech recognizer and screen reader units and performed very satisfactorily.

** Reasonable cost.

Disadvantages:

** The output voice quality, though good, has room for improvement.
The vita has been removed from the scanned document