AFRL-IF-RS-TR-2000-106
Final Technical Report
July 2000

ELSIE: THE QUICK REACTION SPOKEN LANGUAGE TRANSLATOR (QRSLT)

Language Systems, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. D826

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

DTIC QUALITY INSPECTED 4

20001002 042

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2000-106 has been reviewed and is approved for publication.

APPROVED: Sharon M. Walter

SHARON M. WALTER
Project Engineer

FOR THE DIRECTOR: John V. McNamara

JOHN V. McNAMARA, Technical Advisor
Information & Intelligence Exploitation Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFED, 32 Brooks Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

ELSIE: The Quick Reaction Spoken Language Translation

Christine A. Montgomery
David J. Crawford

Contractor: Language Systems, Inc.
Contract Number: F30602-96-2-0196
Effective Date of Contract: 15 May 1996
Contract Expiration Date: 15 October 1998
Program Code Number: D826
Short Title of Work: ELSIE: The Quick Reaction Spoken Language Translation (QRSLT)
Period of Work Covered: May 96 – Oct 98
Principal Investigator: Christine A. Montgomery
Phone: (818) 703-5034 x-10
AFRL Project Engineer: Sharon M. Walter
Phone: (315) 330-7890

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Sharon A. Walter, AFRL/IFED, 32 Brooks Road, Rome, NY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	JULY 2000	Final	May 96 - Oct 98

4. TITLE AND SUBTITLE

ELSIE: THE QUICK REACTION SPOKEN LANGUAGE TRANSLATOR (QRSLT)

5. FUNDING NUMBERS

C - F30602-96-2-0196
PE - 63570E
PR - D826
TA - 00
WU - 01

6. AUTHOR(S)

Christine A. Montgomery and David J. Crawford

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Language Systems, Inc.
5959 Topanga Canyon Blvd, Suite 340
Woodland Hills CA 91367-3648

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

AFRL/IFED
32 Brooks Road
Rome NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2000-106

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Sharon M. Walter/IFED/(315) 330-7890

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this effort was to develop a prototype, hand-held or body-mounted spoken language translator to assist military and law enforcement personnel in interacting with non-English-speaking people. Spoken language translation technology accepts speech input in one human language and generates its computer-spoken translation in another human language. The QRSLT system, nicknamed "ELSIE", demonstrates two-way translation within military and law enforcement vocabularies between English and Spanish, and to a lesser extent between English and Mandarin Chinese and between English and Korean. The system is an innovative advance over currently available "speaking translators" which produce speech based on typed inputs, cannot accept spoken input, and are not customized for military for law enforcement operations. Participation in Global Patriot '98 involved the development and demonstration of capabilities for medical interactions with Korean speakers. Computer-based spoken language translation will have a major impact on soldiers in such areas as: aiding medical personnel in assisting non-English-speaking victims, as a screening aid for interrogating enemy personnel to assist in mine detection, for military foreign language training and proficiency maintenance for interoperability in multi-national operations, and for humanitarian operations.

14. SUBJECT TERMS

Spoken language translation, speech, language processing, speech synthesis

15. NUMBER OF PAGES

122

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1 INTRODUCTION AND SUMMARY....1
1.1 Introduction....1
1.2 QRSLT/ELSIE System Components....1
1.2.1 The Translation Component....3
1.2.2 Speech Processing Components....3
1.3 Overview of System Development Cycles....4
1.4 Summary of Final Technical Report....6
2 CORPUS DEVELOPMENT FOR MILITARY AND LAW ENFORCEMENT....7
2.1 Preparation of Initial Military and Law Enforcement Corpus....7
3 THE TRANSLATION COMPONENT: Translation Strategies and System Implementations....11
3.1 QRSLT/ELSIE Version 0.3 (3 Month Benchmark System)....11
3.2 QRSLT/ELSIE V.0.6 (6 Month Benchmark)....11
3.2.1 Background....12
3.2.2 Choice of Target Hardware and Software Platform....12
3.2.3 Stage One: Building a Prolog-Only System....12
3.2.4 Summary of PC Prolog Porting Experiments....14
3.3 Lexicon Development for QRSLT/ELSIE V.0.9: Addition of a Third Language....15
3.4 QRSLT/ELSIE V.1.0 (12 Month Benchmark System)....16
3.4.1 New Translation Features....17
3.4.2 How the Translation Module Processes Utterances....18
3.4.3 Interim Summary....19
3.5 QRSLT/ELSIE V.1.9: The Dual BNF Engine....20
3.5.1 Backus-Naur Form and Rule-Based Translation....20
3.5.2 A New Entry Method Based on Parallel Syntax Rules....20
3.5.3 Achieving Language Independence Through External Rules....21
3.5.4 Directions for Further Development....22
3.6 Development of More Advanced Translation Capabilities: Experimental QRSLT/ELSIE for English-Chinese Translation....23
4 THE USER INTERFACE FOR SPOKEN TRANSLATION INTERACTIONS....27
4.1 The Initial User Interface (V.0.3 - "Alternate QRSLT/ELSIE")....27
4.1.1 The Dialog Box Interface....27
4.1.2 The "Attention" Context....27
4.2 Overview of QRSLT/ELSIE's User Interface (V.0.6 - V.1.2)....28
4.2.1 The User Interface in Detail....29
4.2.2 "Hands-Free" Operation with Verbal Commands....32
4.3 Changes to the Graphical User Interface for Display of Romanized Chinese....33
4.4 Upgrade of the User Interface: QRSLT/ELSIE V.1.2 - V.1.5....34
4.5 Addition of User-Defined Utterances: QRSLT/ELSIE V.1.8 - V.2.1....34
4.6 Incorporation of Asian Language Scripts into the QRSLT/ELSIE User Interface....36

5 MARKET ANALYSIS AND COMMERCIALIZATION ACTIVITIES....39
5.1 General....39
5.2 Technology Demonstrations and Exhibits: Military Applications....39
5.3 Technology Demonstrations and Exhibits: Law Enforcement....41
5.4 Participation in Technology Expositions and Trade Shows: Other Applications....43
6 SPEECH RECOGNITION COMPONENT....47
6.1 Initial QRSLT V.0.3/QRSLT/ELSIE 1 with Dragon Dictate....47
6.2 Alternate QRSLT/ELSIE with IBM VTAF....47
6.2.1 Description of the API....48
6.2.2 Development of the Recognition Component....48
6.2.3 Using the Recognition Context in the QRSLT/ELSIE Program....48
6.2.4 Initializing and Using the VTAF Functions....49
6.2.5 Programming Considerations: Ownership and Scheduling....49
6.2.6 Performance Analysis of VTAF Speech Recognition....49
6.2.7 Advantages and Disadvantages of VTAF....50
6.3 Evaluation of the 3 Month Benchmark Systems....50
6.4 QRSLT/ELSIE V.0.6....51
6.4.1 Simultaneous Multiple Language Recognition....51
6.4.2 Continuous Recognition With or Without Prompts....51
6.4.3 Technical Issues Encountered....52
6.4.4 Speech Recognition Issues in Spanish....53
6.4.5 Problems with Mixing Synthesized and Recorded Output....53
6.4.6 Changes in Program Structure....53
6.5 QRSLT/ELSIE V.0.9....55
6.5.1 Design Goals....55
6.5.2 Changes to the Speech Output Module....55
6.5.3 Changes to the Context Manager....56
6.5.4 Implementing Low-Level Control of Sound Input....57
6.5.5 Features for Incorporation into the 12 Month Benchmark....63
6.6 QRSLT/ELSIE in Transition from V.0.9 to V.1.2....63
6.6.1 Processing of Sound Buffers....64
6.6.2 Automatic Setting of Input Threshold....64
6.6.3 New Functionality of the "Set Threshold" Command....65
6.6.4 Graphical Indication of Input Level....65
6.6.5 Directions for Further Development....66
6.6.6 Distribution of an 11 Month Benchmark....67
6.6.7 Summary of Speech Recognition Milestones....67
6.6.8 The Search for Phonetic Equivalents....67
6.6.9 Problems Encountered in Adapting ICSS/VTAF for Mandarin....69
6.6.10 Performance of the Mandarin Recognizer....70
6.6.11 Future Directions....70
6.7 Preparing for the Transition from IBM VTAF to HTK (QRSLT/ELSIE V.15)....70
6.7.1 HAPI and SHAPI....70

6.7.2 Rewriting the SpeechRec and ContextMgr Classes 71
6.7.3 Testing Recognition Accuracy 71
6.7.4 Experimentation with Recognition Parameters 72
6.7.5 Monophone versus Triphone Language Models 72
6.7.6 Integrating HAPI into QRSLT/ELSIE 73
6.8 Dual Language Implementation (QRSLT/ELSIE V.18) 73
6.8.1 Multiple Input Languages with IBM VTAF 74
6.8.2 A Multiple-Recognizer Algorithm for Supporting Multiple Input Languages 74
6.8.3 Modifying the Speech Recognition Object to Support Two Recognizers 74
6.8.4 Testing Dual-Language Recognition 75
6.8.5 Analysis of Test Results 76
6.8.6 False Positives and Wrong-Language Recognition 77
6.8.7 Issues Involving Processing Speed 77
6.9 Transition to the IBM Via Voice Recognizer (QRSLT/ELSIE V.2.1) 78
7 SPEECH GENERATION COMPONENT 79
7.1 Background 79
7.2 ETI's Text-to-Speech Technology 80
7.3 System Status for the First Year of Development 82
7.3.1 Work on American English and Mexican Spanish 82
7.4 The Second Year of Development 84
7.4.1 Work on Mandarin Chinese 84
7.4.2 Language Universal Component Enhancements 85
7.4.3 American English Synthesizer 85
7.4.4 Mexican Spanish Synthesizer 86
7.4.5 General Synthesizer Technology Development 87
8 ENTROPIC SPEECH AND LANGUAGE EXPERIMENTS 88
9 REFERENCES 89

APPENDIX A. Sample of Law Enforcement Dialog Corpus 91
APPENDIX B. Samples from the Mandarin Chinese Dialog Corpus 94
APPENDIX C. Samples of the Fresno County Dialog Corpus 96
APPENDIX D. Korean Dialog Corpus Sample and Development Summary for the Global Patriot Exercise 97

1 INTRODUCTION AND SUMMARY

1.1 Introduction

This final technical report describes development of the Quick Reaction Spoken Language Translator (QRSLT), performed for Rome Research Site, Air Force Research Laboratory, and DARPA under a DARPA TRP 95 award (No. F30602-96-2-0196) by a consortium, the QRSLT Development Consortium or QDC, comprised of three small businesses: Language Systems Inc.(LSI), Eloquent Technology, Inc., and Entropic Research Laboratory. The goal of the QRSLT system is to assist military and law enforcement personnel in communicating with persons who speak a foreign language through the exploitation of speech and language processing technology which is at the leading edge of the state-of-the-art. The QRSLT system constructed under this cooperative agreement has been an extremely successful development, as well as a highly popular and visible system throughout the Air Force and in the law enforcement sector. The QRSLT development began with the TRP award in May, 1996, and continued for two years, with an extension for the development of a more robust Korean language capability for the Global Patriot exercise in 1998. A more advanced version of the system is currently under development as a commercial law enforcement product.

The QRSLT concept is based on the Machine-Aided Voice Translation (MAVT) system previously developed for Rome Laboratory under Contract Nos. F30602-93-C-0098 and F30602-90-C-0058. The MAVT research prototype demonstrated voice-to-voice translation from English to Spanish, Arabic, and Russian, and from these languages to English for an initial vocabulary of 100 - 300 words per language. The interlingua-based translation technology developed by LSI for MAVT is summarized in Montgomery et al. 1995, and Stalls et al. 1994, and fully described in Belvin et al.; earlier work is described in Montgomery et al. 1993, and Montgomery et al. 1994.

As opposed to the MAVT system, which runs on a Sun Voyager or SPARCStation under Unix, the QRSLT system is aimed at PC notebooks and handheld or body-mounted PC hardware for use by military and law enforcement agencies. The QRSLT system, nicknamed "ELSIE" from the initials of the system developers, runs under Windows 95 and NT. It is designed for robust processing of spoken inputs in real time. The development has focused on two-way translation of English/Spanish – i.e., both English-to-Spanish and Spanish-to-English -- with a substantial amount of work on Mandarin Chinese, and a lesser effort on Korean.

1.2 QRSLT/ELSIE System Components

Figure 1.1 shows the basic components of the QRSLT/ELSIE voice-to-voice translation system. According to our original development plans, LSI was to be responsible for the translation component, user interface, and system integration, while Eloquent Technology would handle speech synthesis, and Entropic, speech recognition. As it turned out, LSI performed a substantial amount of the speech recognition work on this project, since Entropic's associates at Cambridge Engineering Laboratory were in the process of porting their Unix-based HTK recognizer to the PC Windows environment during most of the

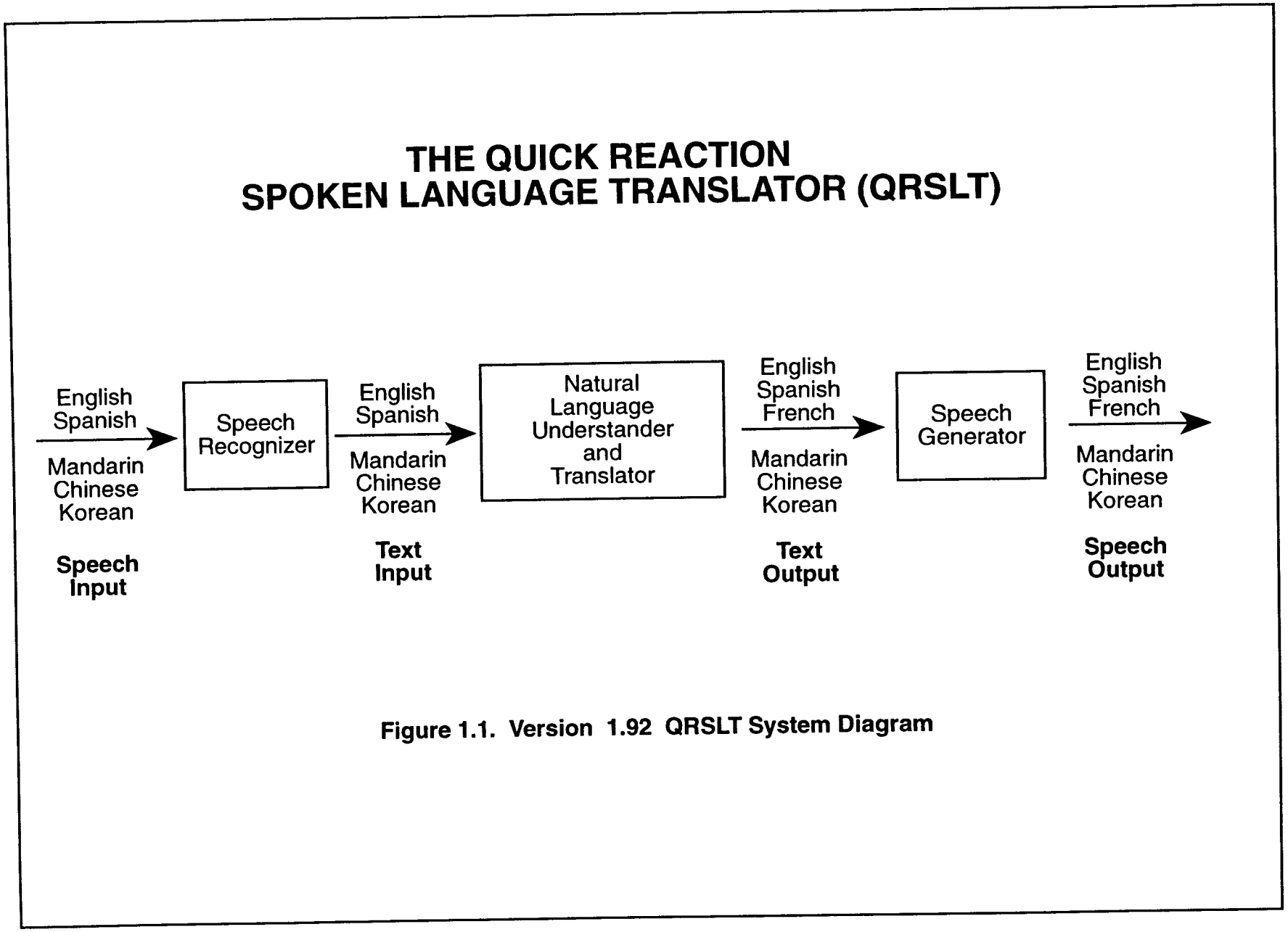


Figure 1.1. Version 1.92 QRSLT System Diagram

QRSLT/ELSIE development, and it was necessary to explore other options in order to maintain the planned development schedule (described in Section 1.3).

1.2.1 The Translation Component

As shown in Figure 1.1, the QRSLT/ELSIE system developed under this Cooperative Agreement translated spoken English to spoken Spanish, Chinese, and Korean, and translated from these languages into English for the dialog corpora developed in the course of the project (described in Section 2).

As noted previously, the original plan for the project was to utilize the prototype of LSI's MAVT system as the basis for the translation component. Since the decision was made by the QRSLT Development Consortium (QDC) partners at the pre-award meeting with DARPA and Rome Laboratory in February, 1996, to develop for the more accessible PC-Windows platform rather than a Sun Unix-based system, plans were made to port the MAVT prototype to the PC platform. Following an extensive test and evaluation of the feasibility and desirability of porting the MAVT system to the PC-Windows environment (Section 3.2), it was decided that this was not a realistic course for the development of a system which was intended to ultimately result in a commercial product. Thus the translation strategies utilized in the QRSLT/ELSIE system are less complex and considerably more efficient than the interlingua-based MAVT translation strategy. The evolution of QRSLT/ELSIE's translation component is discussed in detail in Section 3, which also presents a more advanced MAVT-based experimental system as a foundation for future development.

1.2.2 Speech Processing Components

Speech Recognition

In the course of the project, speech recognizers from IBM and Dragon Systems were utilized in addition to Entropic's English and Spanish recognizers, since Entropic was in the process of porting their HTK recognizer from a Sun Unix platform to the PC-Windows environment throughout most of the development. The initial one-way, English-Spanish spoken translation for the 3 month benchmark system (see 1.3) was achieved using Dragon's speaker-dependent English recognizer. Since the speaker-dependent limitation was undesirable for military or law enforcement applications where respondents to spoken queries would be arbitrary individuals who might be POWs, detainees, or crime victims, LSI acquired a beta version of IBM's Voice-Type Application Factory (VTAF) system, which was used for the alternate 3 month benchmark (Section 6.2) throughout most of the development until the last two quarters, when IBM's commercial product, Via Voice, was substituted. Entropic provided an alpha version of their PC-based recognizers for English and Spanish for the 15 and 18 month benchmarks, which LSI constructed as recognizer-independent systems, since either the IBM VTAF or Entropic recognizers could be used for spoken input (Sections 6.7-8). Due to extensive personnel changes and corporate reorganization within Entropic, later versions of the HTK recognizers were not made available for the QRSLT system;

hence, LSI continued the QRSLT development with the IBM Via Voice recognizer (Section 6.8 and Appendix D).

Speech Generation

Synthesized English, Mexican Spanish, and Mandarin Chinese output is provided via ETI Eloquence, which is described in Section 7.2. Spanish and Chinese speech output can be generated via synthesizer or by using prerecorded wave forms via digital audio playback. Korean is generated only by wave forms. Eloquent Technology also provided the digitized speech for generation of Spanish using wave forms. ETI's TTS system development for QRSLT/ELSIE is described in Section 7.3 and 7.4.

1.3 Overview of System Development Cycles

In an initial pre-award meeting of the QRSLT Development Consortium, when the decision was made to develop for a PC-Windows platform rather than for the Sun Unix platforms on which LSI's MAVT system and Entropic's HTK resided, it was also decided to develop the QRSLT/ELSIE system by a rapid prototyping approach in several cycles, producing increasingly capable systems. The development plan and milestone schedule were thus designed to produce a benchmark system each quarter, as shown in Figure 1.2, with the first year being devoted to development of a two-way voice translation capability for English — Spanish, and the second year to the development and testing of two-way translation for an additional language, originally Arabic. The second language was later changed to Chinese by mutual consent of the Consortium members, and subsequent approval by Rome Laboratory and DARPA. The reason for this change was the more significant commercial value of developing speech components for Chinese, as opposed to Arabic, given the commercialization objectives of the TRP funding.

This rapid prototyping system development strategy is reflected in the descriptions of the major system components below, which discuss the evolution of the QRSLT/ELSIE system in terms of the series of benchmark capabilities constructed through the two-year duration of the project. The systems produced during these cycles can be identified as the initial one-way translation system (v 0.3), the initial two-way system (v 0.6), the initial multilingual (3 languages) system (v 1.2), and the dual-BNF engine system (v 1.8,1.9). Although all the features that were projected to be incorporated in each version were not present in all cases, the aggressive development schedule was maintained by LSI and ETI throughout the project. In some instances, we were able to make earlier than projected introductions of particular capabilities, e.g., LSI delivered a notebook computer containing a version of the QRSLT/ELSIE system that included Chinese to Rome Laboratory after only 7 months of development. Also, we did incorporate a fourth language, Korean, originally as a demonstration capability and subsequently, as a system capability for use at the Global Patriot exercise in 1998.

DEVELOPMENT SCHEDULE AND MILESTONES: QRSLT (YEARS 1&2)

Task	Month*	Year 1 Milestones
1	01/01	Technology Deliveries
1	01/30	HW/SW Inst; Corpus Def.
1,3,4	02/30	QRSLT v.0.3 Desn; Data Coll
1,2,3,4	03/30	QRSLT v.0.3 Integration & Test
1,2,3,4	04/30	QRSLT v.0.6 Design/Developmnt
1,2,3,4	05/30	QRSLT v.0.6 Components
1,2,3,4	06/30	QRSLT v.0.6 Test & Integration
2,3,4	07/30	QRSLT v.0.9 Design/Developmnt
2,3,4	08/30	QRSLT v. 0.9 Components
1,2,3,4	09/30	QRSLT v.0.9 Test & Integration
1,2,3,4,5	10/30	QRSLT Mods, Testg, Imprvments
2,3,4,5	11/30	QRSLT v. 1.0 Components
1,2,3,4,5	12/30	QRSLT v. 1.0 Test & Integration

Task	Month	Year 2 Milestones
1	01/01	Speech Env Maint Delivery
1	01/30	Chinese Corpus Definition
1,3,4	02/30	Chinese QRSLT v.0.3 Components
1,3,4	03/30	QRSLT v.0.3 Integration & Test
1,2,3,4	04/30	QRSLT v.0.6 Design/Development
1,2,3,4	05/30	Chinese QRSLT v.0.6 Components
1,2,3,4	06/30	QRSLT v.0.6 Test & Integration
2,3,4	07/30	QRSLT v.0.9 Frames; Tmp Conc
2,3,4	08/30	QRSLT v. 0.9, 1.0 Components
1,2,3,4	09/30	QRSLT v.0.9 Test & Integration
1,2,3,4,5	10/30	QRSLT Mods, Testg, Imprvments
2,3,4, 5	11/30	QRSLT v. 1.0 Components
1,2,3,4,5	12/30	QRSLT v. 1.0 Test & Integration

Figure 1.2. QRSLT Development Schedule with Benchmark System Versions

*After receipt of contract; effective date was May 14, 1996

1.4 Summary of Final Technical Report

This report is comprised of 9 sections and 4 appendices. The contents can be summarized as follows:

- Section 2 describes development of the various dialog corpora for military, law enforcement, and emergency medical applications;
- Section 3 discusses the evolution of the translation component, evaluation of the feasibility of porting the MAVT system to the PC environment, and development of a series of translation strategies for the various benchmark systems;
- Section 4 describes the evolution of the user interface and associated functions under user control, beginning with a simple dialog box and evolving into a more complex display with greatly improved functionality;
- Section 5 discusses commercialization activities, focusing on market analysis and user requirements definition via technology demonstrations and participation in technology expos and trade shows;
- Section 6 presents a detailed description of the evolution of the speech recognition component, discussing the advantages and disadvantages of the 6 different recognizers used in the QRSLT/ELSIE system at various stages of development;
- Section 7 describes ETI's speech synthesis technology, and details the development of QRSLT/ELSIE's TTS capabilities for English, Mexican Spanish, and Mandarin Chinese;
- Section 8 briefly describes experimentation in which multiple speech recognizers were integrated and evaluated for pairwise language identification and utterance recognition accuracy under varying parameter values;
- Section 9 presents the references;
- Appendix A provides a sample of the initial bilingual dialog corpus (English-Spanish) for law enforcement applications;
- Appendix B gives samples of the Chinese dialog corpora in romanized format (Pinyin) and in traditional Chinese characters;
- Appendix C presents samples from the extended law enforcement dialog corpus obtained from the Fresno County Sheriff's Department;
- Appendix D describes the development of the extended Korean language capability for the Global Patriot exercise.

2 CORPUS DEVELOPMENT FOR MILITARY AND LAW ENFORCEMENT

2.1 Preparation of Initial Military and Law Enforcement Corpus

LSI's primary focus during the first quarter of the QRSLT development was on definition and preparation of the initial bilingual corpus of dialogs for military and law enforcement applications. The initial law enforcement corpus consisted of lexical items and phrases extracted from a law enforcement training course, which was made available to LSI by the Los Angeles County Sheriff's Department.

The bilingual dialog corpus was useful insofar as it provided both the kinds of questions and commands that would be spoken by an officer, as well as an extensive sample of some of the vocabulary that might be used in responses to questions or commands spoken by persons who were detained or arrested: for example, words for family relations, eye and hair color, etc. The full initial law enforcement corpus was presented in Attachment 1 of TRP Milestone Report 2; a representative sample of the types of utterances comprising this corpus is included as Appendix A.

The initial military dialog corpus was defined from the set of expressions listed on the military Command & Control cards prepared by the Defense Language Institute for use by operational units. Command & Control cards provided to us by DLI for Haitian Creole (which has a more extensive collection of words and phrases on a variety of topics than cards for other languages) were translated into Spanish, and made available in several versions to the consortium members and to Rome Laboratory. Examples of DLI Command and Control Cards for Serbian and Croatian were made available to the consortium members, Rome Laboratory, and DARPA.

For the first implementation of the system (V 0.3), a small corpus drawn from both the law enforcement and military sources was used for the purpose of demonstrating the path through the three major system components (i.e. speech recognition, language translation, and speech generation). The English version of the initial dialog sample is presented below to illustrate the types of utterances handled by the first version of the QRSLT/ELSIE system:

Sample Law Enforcement Dialog

Who is the owner of the car?
Where is the car registration?
Give it to me, please
This is a mechanical warning
You should fix your problem and have the correction verified
Take this to an interpreter
This is a citation
Sign here
When you sign the citation, you are not admitting guilt
Go to this court
At this address
On this date
At this hour

In this town

Sample Military Dialog

We are Americans
Lower your hands
Do you speak English?
You are safe
Do not be afraid
We are finished
Stay here
Hello
See you
Thank you
Help will be here soon
We are here to help you
Move back
Form a line
One at a time
Speak slowly
Say it again

In the development of the second version of the system (V. 0.6), a third dialog was added to the two dialogs listed above. This "emergency medical dialog" is shown below:

Emergency Medical Dialog
Are you injured?
Does your chest hurt?
Where does it hurt?
Show me.
You are injured, please do not move!
Are you ill?
Are you a diabetic?
Do you have heart trouble?
How do you feel?
Are you taking medication?
Where is your medicine?
You need medical care.
Do you want a doctor?
Do you want an ambulance?
You should see a doctor.
Do you want to go to the hospital?
You have to go to the hospital.
Where is your medical card?

The query segment of the dialogs was defined initially for English and Spanish. Spanish responses for these queries were developed later, after an initial Spanish recognizer had been

bootstrapped from the IBM VTAF English recognizer (see Section 6.3) In the third quarter of the project, these dialogs were translated into Mandarin Chinese. The Mandarin dialogs were transcribed in both Pinyin (Romanized) and traditional Chinese characters. Samples of these corpora are included with this report as Appendix B.

In addition to these dialogs, a set of verbal commands had to be defined as well (i.e. sentences like *Load English-to-Spanish law enforcement context*). Although specifying this kind of sentence set differs in many ways from the definition of a traditional natural language corpus, it is becoming an important consideration in the development of language processing software with *spoken* language interfaces, as in this project. The contents of the command language are determined by the functionality of the system. Thus, this set evolved from a small list of around 20 commands, as the system developed, and functionality increased. As the number of languages and dialogs the system could handle increased, the number of verbal commands also increased. At the end of the project, the system included a verbal command set of several hundred possible variants. These include full sentences like *Open English to Spanish law enforcement context,* as well as a fairly exhaustive set of abbreviated commands, like *Open police sentences.*

As the project progressed, input from potential user groups was considered and incorporated into the dialog corpora. Most importantly, a second law enforcement dialog corpus was developed around the end of the first year of the project based on a set of jail booking sentences furnished by the Fresno County Sheriff's Department. This corpus is for English-Spanish dialogs only. Samples of the sentences are shown below, while a more extensive sample of the dialog corpus is presented in Appendix C.

Fresno Country Sheriff's Department Booking Questions
What is your nationality?
How much do you weigh?
How tall are you?
What color are your eyes?
What color is your hair?
What is your date of birth?

In addition to the booking forms and other data collected from Fresno, we enlarged the dialog corpora for the law enforcement domain by working with two other organizations, specifically, the Los Angeles County Sheriff's department and a corrections facility in Rhode Island. In both cases, we collected booking sheets (the form used when an arrestee is booked into jail), as well as other questions asked during jail intake, including medical screening questions. The questions asked during booking by the Fresno Sheriff's department had much in common with those asked in LA County and Rhode Island. Some of these questions are illustrated in the sample given in Appendix C from Fresno County. Sample sentences from the medical screening dialog are shown below, and in the more extensive sample in Appendix C.

Fresno County Sheriff's Department Medical Screening Questions
Do you take insulin?
When is last time you took your insulin?

When is the last time you ate?
Have you been drinking?

Because of military interests in spoken Korean translation, the initial dialogs were also translated into Korean. As a final task in the project, additional Korean military dialogs were developed for the Global Patriot exercise in 1998. A Korean speech recognizer was also bootstrapped from the IBM Via Voice English recognizer to allow limited recognition of Korean responses to the English queries. This task development for the Global Patriot exercise is described in Appendix D, which also presents samples of the military dialogs with Korean translations.

In addition to the new dialogs and languages added, all dialogs were further extended to include variants of the sentences in the original set, such that users were not required to read desired sentences directly from the screen, but could utter a given question or command using whatever variant they would ordinarily use, whether it was displayed or the screen or not. Thus, for the following sentence displayed on the screen

What is your home phone number?

the system would also recognize variants such as

What is your home number?
What's your home phone?
What is your phone number at home?
Home phone number?

This effectively increased the number of sentences in the dialog corpus many times over.

Many of the dialogs added later on contained variables, which the system was required to accommodate. For example, the dialog informing a detainee of the date for an appearance in court consists of only a few sentences, but all of these have variable elements for day, date, month, and year, e.g.,

Your court date is <day> <date> <month> <year>

Thus the number of possible utterances for the recognizer to handle becomes extremely large.

Similarly, many of the responses in certain dialogs generate extremely large numbers of possible utterances that the recognizer must distinguish among:

How much do you weigh? Peso <number> kilos | libras
How tall are you? Mido <number> centimétros | <number> pies <number> <fract> pulgadas
How old are you? Tengo <number> | <number> años

In summary, the number of possible utterances -- both queries and responses -- in the final dialog corpus is inevitably quite large.

3 THE TRANSLATION COMPONENT: Translation Strategies and System Implementations

The original intent of our development strategy for QRSLT/ELSIE was to port as much of our earlier research prototype translation system as possible from a Sun UNIX to a PC platform (Microsoft's 32-bit Windows environment). However, our initial efforts in this undertaking, which are described in more detail in Section 3.2, demonstrated to us considerable difficulties which eventually led us to revise our development strategy. Ultimately, we did incorporate some aspects of the MAVT translation system design into the QRSLT/ELSIE system, but almost all of the Sun UNIX implementation proved too cumbersome to port to the desired PC platform, at least if the goal of "near real-time" processing were to be realized. Instead, we traded off depth of linguistic analysis for processing speed, and developed an effective translation strategy based on parallel BNF grammars to achieve the goals of this project. This section of the report details the evolution of this approach through the various versions of the QRSLT/ELSIE system.

3.1 QRSLT/ELSIE Version 0.3 (3 Month Benchmark System)

The primary goal of the initial translation component was to provide an early demonstration of spoken translation in the PC Windows environment. This demonstration integrated a translation component with speech recognition and speech generation components, providing an entire path through the system, beginning with spoken language input in the source language (English), and ending with spoken language output in the target language (Spanish). For the initial system, a simple "direct" translation method (phrase/sentence matching strategy) was adequate. The program was implemented for English to Spanish for the restricted law enforcement and military dialogs presented in the previous section. This initial version of QRSLT/ELSIE demonstrated a one-way path through the system, translating spoken English into spoken Spanish.[1]

The development work involved in QRSLT/ELSIE V. 0.3 was not primarily in the translation component, but in the definition of an initial corpus (see Section 2) and in the integration of the speech recognition, translation and generation components. As noted above, the translation component took source language text as input, and output target language text, using a simple phrase/sentence matching strategy. The source language text was input from the recognition component, and the target text was input to the generation component.

3.2 QRSLT/ELSIE V.0.6 (6 Month Benchmark)

For the 6 month benchmark system, the translation component achieved two-way translation, as noted below. The approach of direct translation based on a phrase/sentence matching strategy now was applied both to English – Spanish and Spanish – English translation, with some improvements to the matching module to increase the speed of translation. Also, in this project

[1] Two-way translation was not introduced until the following (six-month) benchmark, which was based on LSI's "alternate" QRSLT/ELSIE V. 0.3 This system used the IBM VTAF speech recognizer to avoid problems encountered with Dragon Dictate (see Sections 6.1, 6.2, and 6.3 for details).

period, work was begun on attempting to port the MAVT system previously developed by LSI to the PC Windows platform. This work, which continued for some time and eventually resulted in the experimental system for English-Chi n ese translation described in Section 3.6, is discussed in detail below.

3.2.1 Background

LSI's previous translation systems, MAVT (Machine Aided Voice Translation) and MAVT-ADM (Machine Aided Voice Translation-Advanced Development Model) were research prototype systems developed for the Sun SPARCstation under Sun Unix. The translation portion of these systems was written mostly in Quintus Prolog with the Graphical User Interface and a few utility functions written in C++. Since these systems were intended as research prototypes, we had concentrated on building a linguistically powerful parsing and translation engine, but had paid little attention to performance optimization.

In the context of the QRSLT project, the intent has been to build a system that is commercially viable: one that will run on inexpensive hardware and at high enough speeds so that users of the system will not notice a significant delay between voice input and translated (synthesized) spoken output. The twofold challenge has been to move the research prototype system from an engineering workstation to a personal computer hardware platform, and to improve the execution speed of the program by at least an order of magnitude.

3.2.2 Choice of Target Hardware and Software Platform

Based on the state of the computer industry at the time, the choice of a hardware platform for the target system was relatively simple. Microsoft's WIN32 API (Windows 95 and Windows NT) running on microprocessors of the Intel X86 family had become the dominant operating environment for inexpensive personal computers.

The choice of a software development platform was less clear cut. Microsoft's Visual C++ Version 4.0 was chosen for the development of the user interface because of the "Microsoft Foundation Class" (MFC) library that gives programmers ready-to-use templates for creating GUI objects quickly and easily. However, when development began, Quintus did not have a PC version of their Prolog compiler that was interoperable with MSVC++ Version 4.0. It was decided that in the initial stages of the project LSI would use SWI-Prolog, a widely available "shareware" version of Prolog developed at the University of Amsterdam. The advantage of using this version was that the source code for the compiler itself was freely available and could be modified as needed for performance optimization. And since the Prolog compiler's source code was in the C language, it could also be imported seamlessly into a C++ GUI. The disadvantage was that there are minor differences in the implementation of the Prolog programming language between the Quintus and SWI versions. Some Prolog code would require modification to run on the new platform.

3.2.3 Stage One: Building a Prolog-only system

Although the GUI version of the MAVT program is an amalgam of Prolog and C++, the core of the translation system can be separated for test purposes into a version that contains only Prolog code and runs with a command-line interface. The first step in porting was to compile this test version under SWI-Prolog in Windows 95 and debug it to the point that it could produce

translations identical to those produced by the Quintus Prolog version running under UNIX on a Sun Workstation. There were several sources of incompatibility that had to be rectified, including incompatible compiler directives, proprietary language extensions, differences in reserved words, and a critical difference in "assert" Predicates.

3.2.3.1 Comparison of System Performance: Sun versus PC

Once the Prolog portion of the MAVT system was compiled and running under SWI-Prolog on the PC, the next step was to do some benchmark testing. We wanted to know how SWI-Prolog on a 133 MHz Pentium PC compared in performance to Quintus Prolog running on a Sun SPARC 5, and we also wanted to find out how much processing time was being used by the various stages of the translation process. The following tables present the results of running several sample sentences on both platforms:

Test Sentence: The infantry attacked the bunkers.

Translation: *La infantería atacó los búnquers.*

Processing time (in CPU milliseconds):

Processing Step	Quintus Prolog on Sun SPARC 5	SWI-Prolog on 133 MHz Pentium
Lexicalization	130	60
Syntactic Parse	1010	550
Functional Parse	1960	1260
Template Creation	1360	710
Generative Functional Parse	480	210
Generative Parse	620	390
Total Processing Time	**5560**	**3180**

Test Sentence: Did the ambassador go to the hospital?

Translation: *¿Fue el embajador al hospital?*

Processing time (in CPU milliseconds):

Processing Step	Quintus Prolog on Sun SPARC 5	SWI-Prolog on 133 MHz Pentium
Lexicalization	160	110
Syntactic Parse	5420	3840
Functional Parse	3200	1810
Template Creation	1430	700
Generative Functional Parse	540	170
Generative Parse	740	440
Total Processing Time	**11490**	**7070**

Test Sentence: The troops ran across the beach.

Translation: *Las tropas corrieron a través de la playa*

Processing time (in CPU milliseconds):

Processing Step	Quintus Prolog on Sun SPARC 5	SWI-Prolog on 133 MHz Pentium
Lexicalization	140	110
Syntactic Parse	14540	12240
Functional Parse	2410	1530
Template Creation	1420	620
Generative Functional Parse	540	320
Generative Parse	730	440
Total Processing Time	**19780**	**15260**

It appears that the combination of SWI-Prolog and the 133 MHz Pentium Processor is somewhat faster than Quintus prolog running on the Sun SPARC 5. Having no reason to believe that a non-commercial implementation of Prolog is any faster than Quintus Prolog, which is probably the most widely used commercial Prolog compiler, we would tend to attribute this performance gain to the underlying speed of the CPU.

3.2.4 Summary of PC Prolog Porting Experiments

Overall, we had to concede that in spite of the modest performance gain seen on the PC, the overall speed of translation was still far too slow to meet the needs of a conversational speed translation system. Although there are ways to improve the performance demonstrated in these benchmark tests, (such as rewriting key Prolog predicates in C/C++), it is unlikely that the combined performance improvements we might make would have been enough to give the order of magnitude increase in execution speed that would be required for a conversational speed translation system. In the end it was determined that it would be necessary to rewrite entire sections of the program in C/C++ and interface them with whatever portion of the system remained in Prolog.

In any programming project there are likely to be tradeoffs between power and performance. If all optimization possibilities have been exhausted and the performance still is not adequate, the designers of a system have no choice but to simplify their program by removing features or choosing simpler algorithms.

An important factor we were considering in our design of a conversational-speed spoken language translation system was that the set of possible sentences to be translated would be limited by the capabilities of the speech recognizer. Even the best speaker-independent, continuous speech recognizers must be restricted to a relatively small vocabulary to achieve adequate accuracy in the type of application we are building.

Another factor which figured in the design was that the QRSLT system is meant to be used in well-defined applications such as military intelligence, law enforcement, or emergency medicine. The syntactic and semantic parsing which take up the bulk of processing time can be greatly simplified by narrowing the scope of the parser to the most critical commands and questions required in those applications. In order to determine what type of syntactic and semantic structures characterized such sentences, we collected actual utterances and example sentences from several different law enforcement agencies (LASD, Fresno SD, a Rhode Island corrections facility. For military dialogs, we utilized the Command and Control cards, generating response contexts for these as required.

3.3 Lexicon Development for QRSLT/ELSIE V.0.9: Addition of a Third Language

In the earlier versions of QRSLT/ELSIE, sentences were treated as neutral sequences of characters which were processed as a unit. The lexicon of the Sentence Translation module did not distinguish between the sentences that were in English and those that were in Spanish. Each sentence was stripped of punctuation and then stored with a pointer to another sentence identified as its translation. (The stripping of punctuation was necessary because the raw input from the Speech Recognition module contains no punctuation; if a sentence match is to be found, the stored sentence against which the recognizer input is compared must be in the same format.) If a recognized sentence could be found in the list of stored sentences, its translated version would be returned and subsequently spoken by the Speech Output module.

In Version 0.6 the lookup method was made more efficient by the addition of a hash table so that looking up the last sentence in the list did not require performing a string matching function on every single sentence, but the information that was stored was essentially the same.

With the addition of a third language, this is no longer possible. The same sentence in English may appear twice in the database, once with a Spanish translation and once with a Mandarin translation. The program needs to be able to distinguish between the two.

While we were redesigning the lexicon, we tried to make it flexible enough to hold individual words or phrases as well as complete sentences. This decision was motivated by the fact that we were planning to enhance the program's translation capabilities and needed to have a lexicon for storing individual words. We decided that it would be most efficient to have all lexical information stored in the same location.

The following is the LexEntry structure format for the new lexicon:

```
typedef struct LexEntrytag {
        char *str;
        char Category;
        char *base_str;
        char *trans_str;
        char src_lang;
        char tgt_lang;
        int WordValue;
        int Attributes;
        LexEntrytag *NextNode;
        LexEntrytag *NextBaseNode;
} LexEntry;
```

The character string "str" stores the word (or sentence) in question, and "Category" is an enumerated type indicating the lexical category, or part of speech (LSI_NOUN, LSI_VERB, etc.). In the case of a complete sentence, a special category LSI_UTTERANCE is used. (The "LSI_" prefix is used on all enumerated types for the lexicon to avoid namespace problems.) The "base_str" performs different functions for words and sentences. For a word, it is the "root" or "base" form of the word (e.g. the root form of the word "is" is the word "be"). For a sentence, the "base_str" represents the sentence stripped of punctuation. The "trans_str" is the translation of the word or sentence. Two more enumerated types, "src_lang" and "tgt_lang", identify the languages of "str" and its translation. "WordValue" is an integer value applied to words; it is not used at present, but will eventually help the parser distinguish between common and uncommon meanings of the same word. "Attributes" is a bit-field containing information about a word's attributes (gender, number, tense, etc.). The last two structure members, "NextNode" and "NextBaseNode", point to the next node in each of two hash chains. There are two separate hash tables, so each entry can be looked up either by its inflected form or by its base form. In the case of complete sentences, the base form (unpunctuated) is used for lexical lookup while the normal form (with punctuation) is used for display by the graphical user interface.

When the program attempts to look up a sentence, it calls a member function from the "SentenceTrans" class that has the following prototype:

```
LexEntry *LookUpBaseEntry(char *base_str, char src_lang, char tgt_lang);
```

So, for example, the function call:

```
LookUpBaseEntry("this is a test sentence", LSI_ENGLISH, LSI_MANDARIN);
```

would either return a pointer to the LexEntry containing the Mandarin translation of that English sentence in its "trans_str" member or else return NULL if the sentence was not in the lexicon.

3.4 QRSLT/ELSIE Version 1.0 (12 month benchmark)

For the 12-month benchmark version of the program to be demonstrated and distributed to the consortium members at the One-Year DARPA Review Meeting, a new version of Eloquent's

English and Spanish text-to-speech engine was incorporated into the program, and several features were added to the translation module to increase its power and flexibility.

During this quarter, the existing Law Enforcement and Medical dialog contexts were extended to be fully bi-directional in both English-Spanish and English-Mandarin versions. In addition, as mentioned in the preceding section, a new English-to-Spanish corpus based on a jail booking dialog provided by the Fresno County Sheriff's Department was implemented. The challenge of adapting QRSLT/ELSIE to handle these utterances from an actual set of questions and statements used by correctional officers in Fresno led us to extend the program's translation capabilities in several directions.

3.4.1 New Translation Features

In prior benchmark releases of QRSLT/ELSIE, translation between all languages recognized by the program (English, Spanish and Mandarin) was accomplished by a direct transfer strategy, based on looking up entire utterances in a table. LSI had been working for some time on more flexible translation procedures, but these had been used only in experimental versions of the program. For the 12-month benchmark, we felt that several of these translation procedures were now robust enough to be included in the official version of the program.

In addition to simple direct transfer of entire utterances, the following translation methods were incorporated:

Alternative Forms for Existing Utterances – To allow for commonly used alternatives to sentences already included in the various dialogs, the ability to enter alternative forms was added to the program. For example, the following sentence-translation pairs are from the "booking" section of the Fresno Jail corpus:

What is the [number][2] of your [apartment | house]

¿Cuál es el [número] de su [apartamento | casa]?

What is your [apartment | house] [number]?

¿Cuál es el [número] de su [apartamento | casa]?

These alternative forms are still looked up as complete utterances, but their presence extends the flexibility of the system by covering additional variations on sentences in the dialog corpora. Users of the system thus do not need to read questions from the display, but can use whatever alternative is most convenient to them in a given dialog; this is another step towards a completely hands-free system.

Compound Utterances – In real-world tests of QRSLT/ELSIE with non-technical users, it was found that users would often say more than one utterance at a time. This was particularly true for short utterances covering the same or a related subject that seemed to follow one another logically. For example, the Law Enforcement dialog contains the utterances "Do not be afraid" and "You are safe," which could easily be combined into a single utterance. Conversely, other utterances in the existing corpora seem to consist of more than one utterance already. A good example of this from the Medical Corpus is the utterance "You are injured, please do not move." Either clause from this utterance could be (and in test situations frequently was) spoken by itself.

[2] The words in square brackets are the key words for the sentence, as described in Section 3.4.2.

The ability to string together multiple utterances was particularly useful in the newly implemented Fresno Jail corpus. One entire section of this corpus began with the utterance "The crime for which you were arrested is...", followed by a list of two dozen separate charges that might complete that sentence.

In order to accommodate these compound utterances, a new translation strategy was introduced, specifying a function in the translation module that attempts to match a recognized string of words with multiple entries from the table of known utterances.

Syntactic Parsing and "True" Translation – An experimental Natural Language Processing module that parses and translates from English to Spanish was introduced into this version of the program in order to handle a specific dialog in the Fresno Jail corpus. One of the sentences submitted to us was "Your court date is January 2, 1997." Clearly this utterance is only useful if other dates can be substituted. As a first test of our natural language parser, we made up a set of rules for recognizing and translating dates in several different formats. The following sample sentence patterns will all be recognized and correctly translated:

Your court date is January 2nd.

Your court date is February 17th, 1998.

Your court date is the 27th of August, 1997.

March 4th is your court date.

September 12th, 1997, is your court date.

The 19th of July, 1997, is your court date.

The translation will work the same way either for cardinal numbers (e.g. one, two, three) or ordinal numbers (e.g. first, second, third), and will be rendered accurately in Spanish in the cardinal form.

Since the translation is done through syntactic parsing, not table lookup, any noun phrase that the program can translate could be substituted for "your court date" in any of the sentence patterns shown above. For example, the sentence "The date is January 2nd" or "Your day in court is January 2nd" would both be translated correctly.

In practice, however, the latter sentences would have to be entered into the speech recognizer's grammatical context in order for them to be recognized and submitted to the translation module. In order to demonstrate the abilities of the translation module independently of the recognizer, a new feature was added to QRSLT/ELSIE's "manual" mode of operation in which the user could directly type in a sentence to be translated instead of speaking the sentence into the microphone.

3.4.2 How the Translation Module Processes Utterances

The translation module of QRSLT/ELSIE has several different techniques that can be applied to achieve translation. The following is an algorithmic description of the steps involved in processing a string of words submitted by the speech recognizer.

1) **Look up the entire string in the dictionary** – If the whole string has been entered as a complete utterance in the program's lexicon, direct translation and output (recorded or synthesized speech) can take place immediately.

You have the right to remain silent.

Usted tiene el derecho de permanecer callado.

2) **Attempt to split the string into multiple known utterances** – If lookup of the whole string fails, the translation module checks to see if the string can be segmented into multiple complete utterances. The procedure is recursive, so any number of utterances could be combined. If the string is split successfully, each component utterance is translated and output separately.

The crime for which you were arrested is possession of drugs.

El crimen por el que fue arrestado es posesión de drogas.

3) **Select key words from the string and look for a match** – If the previous two steps fail, the string will be scanned for words identified as key words in the lexicon. If key words are found, they are combined into a key-word string and looked up. If the key-word string is found in the dictionary, a complete utterance is substituted for the input utterance and processing proceeds as in step 1 above.

Input to recognizer: What is the [apartment] [number]?

Recognized items in input string: [apartment] [number]

Output generated: *¿Cuál es el [número] de su [apartamento]?*

4) **Submit the string for syntactic parsing** – If all the steps above fail, the string is submitted to the syntactic parser. If the string can be parsed, the translation module looks for rewrite rules that can be applied to render the nodes of the parse tree in Spanish. If a valid set of rewrite rules is found, the sentence is translated and sent to the speech output module for synthesis. (There will be no recorded version of the output, since the sentence was not found in the lexicon during the first stage of translation processing.)

Your court date is August 2nd, 2000.

Su fecha en la corte es el dos de agosto de dos mil.

3.4.3 Interim Summary

The system of key-word extrapolation described above is highly dependent on the speech recognizer's grammatical context, or speech grammar. The major limitation of key-word matching at this time is that recognizer errors will sometimes garble the key words in an attempt to match an unknown utterance against a known pattern in the grammar. The effectiveness of key-word matching can be improved by creating new speech grammars that take the key words into account and are more tolerant of variation in the other words in the sentence. The ability to examine "N-best" hypotheses for given words would also be helpful in this translation strategy.

The syntactic parser and translation rewrite rules implemented in this version work well for the limited domain in which they are being used, but they are language specific and for this version of the system, only allow translation from English to Spanish. The next stage in the development of this part of the program is to create a semantic parse after the initial syntactic parse, and from this an interlingual representation and a set of generative rules that can translate the interlingual form into any of the target languages of the system, as in the MAVT prototype. An experimental implementation of the next stage of development is described in Section 3.6.

3.5 QRSLT/ELSIE (V.1.9) – The Dual BNF Engine

In the early stages of the QRSLT/ELSIE project, LSI did port the translation module of MAVT to the WIN32 environment (see Section 3.2). Predictably, the Prolog code on the PC exhibited the same slowness that it had on UNIX workstations. Attempts were made to optimize execution speed by rewriting portions of the Prolog code in the C programming language, but at best these optimizations improved performance by only a factor of two or three. In many settings this level of improvement would be considered substantial, but for a "Quick Response" voice translation system the reduction in execution time from one minute down to twenty seconds was not adequate.

At this time two important design decisions were made: the PC version of the translation system would not contain any Prolog code, and the depth of linguistic analysis would have to be reduced.

3.5.1 Backus-Naur Form and Rule-Based Translation

The key to improving the efficiency of the translation module was the recognition that it needed to perform its task within a limited domain: the set of utterances that could be recognized by the speech recognition system. Since Backus-Naur Form (BNF) syntax rules were used by the speech recognizer (as in most current recognition systems) to create a Finite State Grammar specifying the utterances that could be recognized, those same rules could be used to specify what can be translated.

What we needed, then, was an extension of the Backus-Naur Form that would allow us to specify not only the set of legal syntactic constructions within our grammar, but for each such construction enumerate the legal transformations from the source language to the target language. If the process of translation can be reduced to a set of syntax rules and transformations, the program can translate very quickly no matter how large the database of rules becomes.

3.5.2 A New Entry Method Based on Parallel Syntax Rules

The extension of BNF that we developed places the syntax rules for the source and target language in parallel in the same rule with a separator between them. For example, in the classical BNF used to program IBM's ICSS/VTAF speech recognizer, an entire sentence might be specified as follows:

<sentence> ::= who is the owner of the car .

In LSI's parallel syntax format for translation from English to Spanish, this would be written:

<sentence> ::= who is the owner of the car ^ quién es el dueño del carro .

The example above specifies that the entire sentence *"Who is the owner of the car?"* should be translated as *"¿Quién es el dueño del carro?"*. This shows the parallelism of the source and target languages but does not illustrate the power of BNF notation. To go beyond simple

recognition and translation of whole sentences, BNF constructions must be nested. For example, take the following two rules in ordinary BNF format:

<inquiry_target> ::= the owner of the car | your friend | in the room .

<sentence> ::= who is <inquiry_target> .

Taken together, these two BNF rules specify three legal sentences in our grammar: *"Who is the owner of the car?"*, *"Who is your friend?"* and *"Who is in the room?"*. The vertical bars indicate alternative constructions in a rule. A rule such as <inquiry_target>, when used in the body of another rule, is called a "non-terminal" because it must undergo further expansion before an application of the rule terminates.

In LSI's parallel BNF notation for translation from English to Spanish, the specification for the Spanish translation is added to extend the rules. In addition to being a powerful method for creating translation rules, this notation solved the problem of efficiently creating new content for the QRSLT/ELSIE system. Previously, it was necessary to create three separate files containing standard BNF rules for the speech recognizer, lists of legal sentence types in the source language for display to the user, and pairs of source and target utterances for the translation module. These three files then had to be kept synchronized if any of them were modified. With the new notation, there is a single point of maintenance; only the parallel BNF file needs to be modified if changes are made. The files used as input to the recognizer, user interface, and translator are generated automatically with Perl scripts from the parallel BNF file.

3.5.3 Achieving Language Independence through External Rules

Once we had settled on a notation for representing our translation rules, we needed to create a translation engine that would use those rules. The translation module that was used in the previous versions of QRSLT/ELSIE already used recursive rule-based translation techniques, but the rules and the lexicon were embedded within the program code. In constructing the upgrade to the engine we removed the rules from the program and instead read them in from external files at run-time.

This separation of data from executable code, which is always desirable in computer programming, is similar to the approach that Eloquent has taken in the newest version of their Text-To-Speech system. The end result for LSI's translation engine, as for Eloquent's TTS engine, is language independence. The engine is carrying out rules and does not care what language is being translated as long as it follows those rules.

We made immediate use of this capability in QRSLT/ELSIE by creating an internal "command" language for handling the verbal commands that QRSLT/ELSIE can recognize and act on. Previously, we had hard-coded these commands into the program code. Now, they are kept in a parallel BNF file and can be changed without recompiling the entire program.

Here, as an example, are the rules for specifying the verbal command that a user can give to end the program:

<end_word> ::= end | quit | exit | stop .

<sentence> ::= <end_word> program ^ confirm ack end program .

These two rules specify that any of the four commands *"end program"*, *"quit program"*, *"exit program"* or *"stop program"* will be translated as "confirm ack end program", which tells the user interface that it should prompt the user for confirmation, acknowledge the confirmation when it is given, and then end the program. Notice that translations for the individual words in the <end_word> non-terminal were not necessary because they do not appear in the target of the <sentence> rule.

3.5.4 Directions for Further Development

In the current version of QRSLT/ELSIE the translation engine is being used mostly for the translation of entire sentences or sentences containing one or two non-terminals. In other words, translation is taking place at the sentence or phrase level. However, the engine is scaleable and can be used to perform translation or linguistic analysis at the word level if appropriate rules are input. Thus, the engine could be used as a tool for the development of a more sophisticated system that can accept arbitrary input.

Here, for example, are a few simple rules to allow the engine to translate *"Who is the owner of the car?"* word by word instead of phrase by phrase:

<wh_pronoun> ::= who ^ quién .
<verb> ::= is ^ es .
<determiner> ::= the ^ el .
<noun> ::= owner ^ dueño | car ^ carro .
<prep> ::= of ^ de .

<prep_phrase> ::= <prep> <noun_phrase> .
<noun_phrase> ::= <determiner> <noun> | <noun_phrase> <prep_phrase> .

<sentence> ::= <wh_pronoun> <verb> <noun_phrase> .

These parallel BNF rules specify six legal sentences (not all of which are semantically acceptable) and their translations:

Who is the owner -> Quién es el dueño
Who is the car -> Quién es el carro
Who is the owner of the car -> Quién es el dueño de el carro
Who is the owner of the owner -> Quién es el dueño de el dueño
Who is the car of the owner -> Quién es el carro de el dueño
Who is the car of the car -> Quién es el carro de el carro

A few extra explanatory comments are in order here. The example above has been chosen for the sake of simplicity in that the nouns are both singular and masculine; a more realistic example would have to deal with Spanish morphology as well as the contraction of *"de el"* into *"del"*. In practice, it may be easier to handle these items through post-processing of the Spanish output. Also, one of the rules (<noun_phrase>) is recursive; the translation engine will accept this, but, as is well known, such rules can lead to infinite recursion.

By introducing lexically constrained non-terminals on the right-hand side of the <noun> rewrite rule, e.g., <noun_human>, <noun_object>, adding other similarly specified non-terminals into the rules, and using a lexicon to define the lexical items and their syntactic and semantic features, the simple BNF grammar can evolve toward a semantic grammar that would cover a large number of possible input utterances and specify their translations.

Even so, such a grammar would be unable to deal efficiently with structures that are not parallel in the source and target languages, such as "I like red wine" – "Me gusta el vino tinto": literally, *"Red wine is pleasing to me")*. To handle such structures more effectively, an advanced translation capability is required, such as the translation engine of the MAVT ADM system. Thus, although our attempts to port the MAVT system to the PC directly had not been productive (Section 3.2), we began the development of an experimental translation system based on that model, for future incorporation into QRSLT/ELSIE.

3.6 Development of More Advanced Translation Capabilities: Experimental QRSLT/ELSIE for English – Chinese Translation

Work was carried out to port some of the important features of MAVT technology to an experimental version of QRSLT/ELSIE, which focused on English-Mandarin Chinese translation. In MAVT, we adopted an interlingual approach to translation, i.e., after the morphological and syntactic analysis of the source language sentences, before generation of the corresponding target language structures, we first derive a semantic representation of the source language sentences, expressed in a language independent way (interlingua). The interlingua eliminates the need to construct translation rules for each source-target language pair.

Similar to the MAVT system, the experimental system now processes a source language sentence in several stages. It first parses the sentence and produces a binary parse tree, carrying along any morphological information from our lexicon:

```
DEC_CLAUSE:
 NP:
  NP1:
   DET: the   sing pl
   NP2:
    NP3:
     NOUN: driver   sing
 IBAR:
  VP_FIN:
   V_FIN: signed   sing pl 1stSing 2ndSing 1stPlur 2ndPlur 3rdPlur past base_verb
   RVP:
    NP:
```

```
      NP1:
        POSPRON: his
        NP2:
          NP3:
            NOUN: name   sing
  PP:
    PREP: on
    NP:
      NP1:
        NP2:
          QUANT: three
          NP3:
            NOUN: forms   pl
```

Next, we generate a functional parse from the syntactic parse, which specifies the semantic functions of the various sentence components in the syntactic parse:

```
Clause:
  Predicate:  write ACT PAST PRET IND
  Subject:
    Entity:  driver THIRD SG
    Determiner:  the
  DObject:
    Entity:  name THIRD SG
    PossPhrase:
      Entity:  he THIRD SG POS
  OblPhrase:
    Prep:  on
    OblObject:
      Entity:  forms THIRD PL
      QuantPhrase:
        Quant:  three
```

From this functional parse, we derive the interlingual representation of this sentence. The interlingual representation is implemented with template structures, as in the MAVT system:

```
Event:
Sentence Type: Declarative
Nucleus: SIGN1
Voice: ACT
Tense: PAST
Aspect: PRET
Mood: IND
Agent:
  Nucleus: DRIVER1
```

```
    Person: THIRD
    Number: SG
    Definiteness: DEFINITE
  Patient:
    Nucleus: NAME1
    Person: THIRD
    Number: SG
    Possessor:
      Nucleus: HE1
      Person: THIRD
      Number: SG
      Case: POS
  Location:
    Nucleus: ON1
    Loc:
      Nucleus: FORM1
      Person: THIRD
      Number: PL
      Quantity:
        Nucleus: THREE
```

From this interlingual representation, the system then generates the following functional parse structure for the target sentence in Mandarin:

```
Clause:
  Predicate: qian1 ACT PAST PERF IND
  Subject:
    Entity: si1ji1 THIRD SG
  DObject:
    Entity: ming2zi4 THIRD SG
    PossPhrase:
      Entity: ta1 THIRD SG POS
  OblPhrase:
    Prep: zai4...shang4
    OblObject:
      Entity: biao3ge2 THIRD PL
      QuantPhrase:
        Quant: san1
```

Finally, the surface string for the target sentence is generated:

Si1ji1 zai4 san1 zhang1 biao3ge2 shang4 qian1 le ming2zi4.

The syntactic structures currently being handled by the interlingua component of the experimental system include the following:

Declarative sentences:

- major tense/aspect combinations
- negation on the main verb
- adverbials
- some prepositional phrases including those expressing location and time
- determiners, quantifiers, adjectives, possessives in complex noun phrases

Imperatives:

Same structural variations as in declarative sentences

Questions:

Yes-no questions, with the same structural variations for declarative sentences

As noted previously, the experimental system has been using English as the source language and Mandarin as the target language, in order to test the system in situations where the source language and target language differ in major ways. We have developed a module to handle translation divergences between the source language and the target language. Currently, the divergent structures specific to Mandarin which are handled by this module include the following:

Word order
Classifiers for nouns
Specification of aspect markers based on the tense/aspect values in the interlingua
Distinction between stative vs. conditional sentences
Distribution of copular verb

4 THE USER INTERFACE FOR SPOKEN TRANSLATION INTERACTIONS

With each of the system development cycles described in the preceding section and in Section 6, the user interface for QRSLT/ELSIE was modified to reflect the added functionality and flexibility. This section describes the evolution of the graphical user interface, from a prompt-driven, simple interface to an interface with flexible access to QRSLT/ELSIE's capabilities and modes of use, which accommodates both the roman alphabet and Asian language displays. All user interface development was carried out by LSI.

4.1 The Initial User Interface (V.0.3 – "Alternate QRSLT/ELSIE")

For its first implementation of an alternate (see Sections 1.2 and 6.1, 6.2) 3 month benchmark version of QRSLT/ELSIE using speaker-independent, continuous speech technology, LSI designed a "dialog box" user interface with Visual C++ and the MS Foundation Class library, using IBM's Voice-Type Application Factory (VTAF – based on IBM ICSS—IBM Continuous Speech Series) API in an unreleased version dated May, 1996) for both speech recognition and generation. The .WAV files used in speech generation were provided by Eloquent Technology, Inc.

4.1.1 The Dialog Box Interface

The user interface of the initial QRSLT/ELSIE system looked much like the first display in Section 4.2, except that there was no function for context management, so the "Context Selection" label and the "Load Contexts" button were absent. To operate the system, the "Start" button was pressed; to stop the recognition and translation process, the "Stop" button was pressed. To use Manual rather than Automatic translation mode, the "Settings" button was pressed, bringing up the "Settings" display. This display, at that point, did not have most of the functionality shown in the "Settings" display presented below under Section 4.2.1(4), but did allow the selection of Manual Mode, which operated as described below.

4.1.2 The "Attention" Context

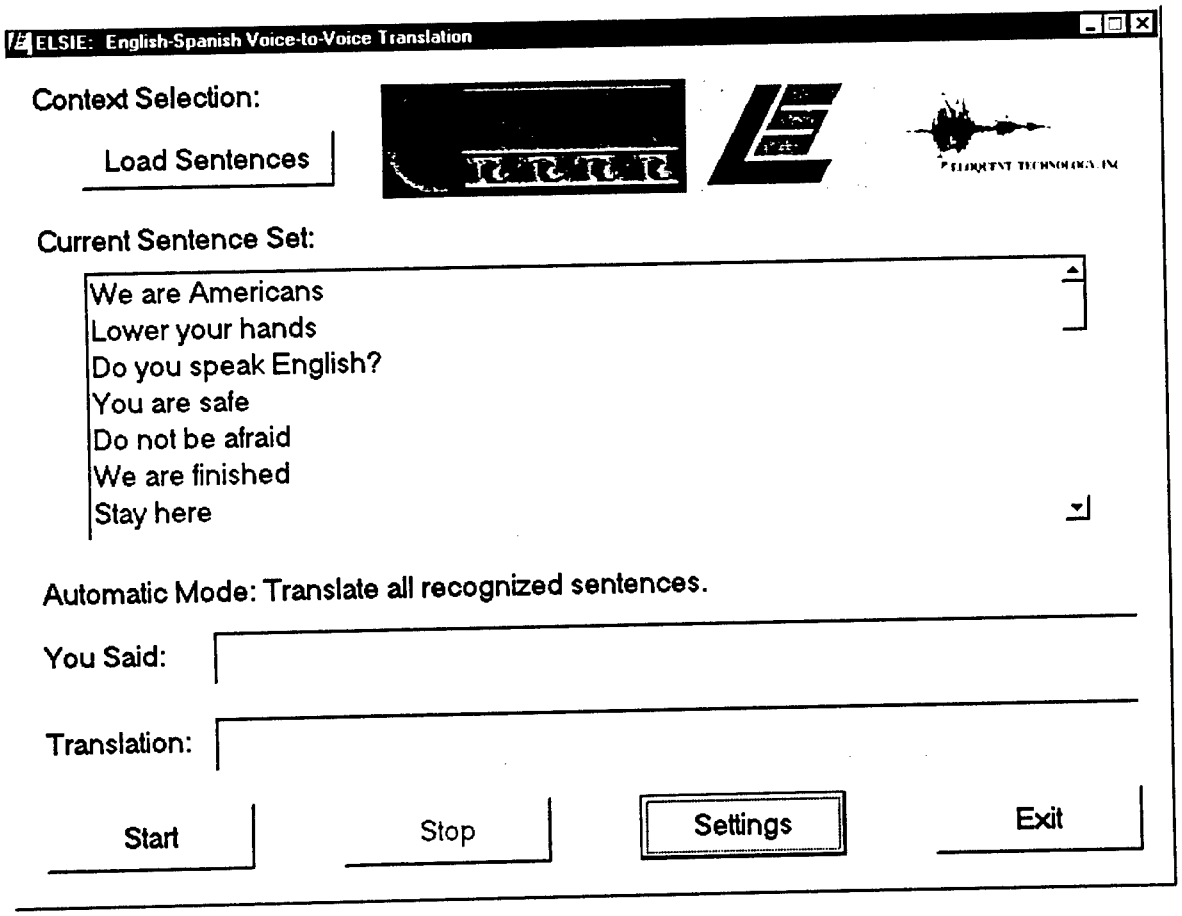
In addition to the speech recognition context, or speech grammar for the initial system, a one-word "attention" context was created with the name "ELSIE". The attention context allowed the program to wait indefinitely for spoken input until it detected the attention word, allowing hands-free operation. When the attention word was spoken, the system responded with one of a number of recorded responses representing ELSIE's feedback to the user, encouraging the user to speak to the system: e.g., "Speak now", "I am here", etc. The program then changed to the sentence context and began listening for content input. The recognition within this normal context timed out after a (configurable) period of silence. If a known sentence was heard during the input period, the program immediately looked up the translation and generated the appropriate .WAV file. As noted previously, the program also included an optional manual mode of operation in which spoken input and subsequent translation and .WAV output can be triggered by sequentially pushing buttons on the dialog box.

Subsequently, the original "attention" context was modified by the addition of a large "SPEAK NOW" display, since we found that users would still hesitate, and the content recognition program would often time out before the user uttered any spoken input. This would of course return the user to the "attention" context, necessitating the repetition of the attention word before initiating a content utterance. This was confusing for many users, while others would begin confidently with the attention word, then utter an initial sentence from the display, and proceed to the next sentence without repeating the attention word, causing recognition errors. We thus decided to discontinue use of the "attention" context in subsequent versions of the system.

It should be noted here that our participation in conferences and technology expositions, described in Section 5, was invaluable for receiving this sort of feedback from potential users and for determining system requirements, both for content and functionality.

4.2 Overview of QRSLT/ELSIE's User Interface (V.0.6 – V.1.2)

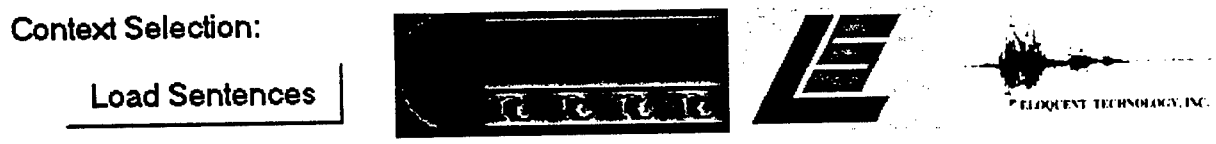
This section presents a detailed description of the functionality of the main features of QRSLT/ELSIE's user interface which were available throughout the development period. Additional features incorporated later are discussed in the following sections, but the functionality described below was retained. Our detailed description begins with the initial screen display of the system.



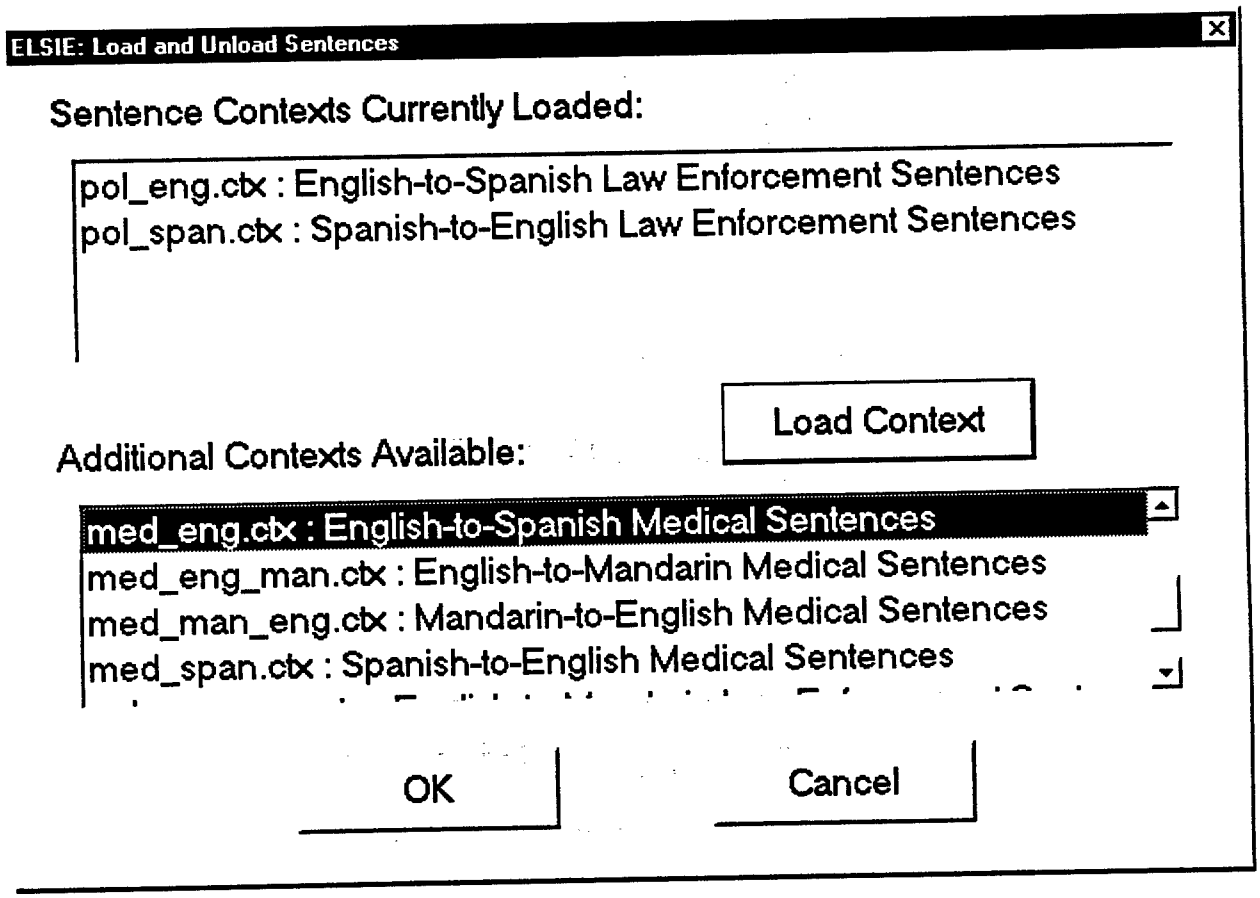
When the user first selects the program by double-clicking on the QRSLT/ELSIE icon, the display on the preceding page appears. In the next section we will examine each part of this screen. QRSLT/ELSIE's modes of operation and the function of each button and box will be explained. Following that, the verbal commands for controlling QRSLT/ELSIE are enumerated, and issues relating to control of sound input levels are discussed.

4.2.1 The User Interface in Detail

1) Context Selection -- In the upper part of the screen, next to the logos for consortium members Entropic Research Laboratory, Language Systems Inc., and Eloquent Technology, there is a button labeled "Load Sentences":



Pushing this button brings up a dialog box that allows the user to enable or disable individual dialog contexts:



The dialog contexts contain lists of sentences that QRSLT/ELSIE can recognize and translate. In the dialog box shown above the English-to-Spanish and Spanish-to-English Law Enforcement contexts have already been loaded. The English-to-Spanish Medical context is highlighted in the lower list; pressing the "Load Context" button will move it to the upper list and enable it.

Clicking with the mouse on any context name will highlight it. If a context in the upper list is highlighted, the caption on the button will change to "Unload Context" and pressing it will deselect the context and move it to the lower list box.

Pressing the "OK" button finalizes the selections, while pressing the "Cancel" button discards any changes made in this dialog box.

Dialog contexts may also be loaded and unloaded using verbal commands. A complete list of verbal commands recognized by QRSLT/ELSIE appears later in this document.

2) List of Sentences – The next part of the screen contains a list of phrases and sentences in the recognition contexts that are currently loaded:

Current Sentence Set:

We are Americans
Lower your hands
Do you speak English?
You are safe
Do not be afraid
We are finished
Stay here

At the right side of the list box is a scroll bar for navigating the list. The verbal commands "Page Up" and Page Down" also allow the user to move up and down in the list of sentences.

3) Input Sentences and Translations – Below the list of sentences are two "Edit Boxes" for the input and output of individual sentences:

Automatic Mode: Translate all recognized sentences.

You Said:

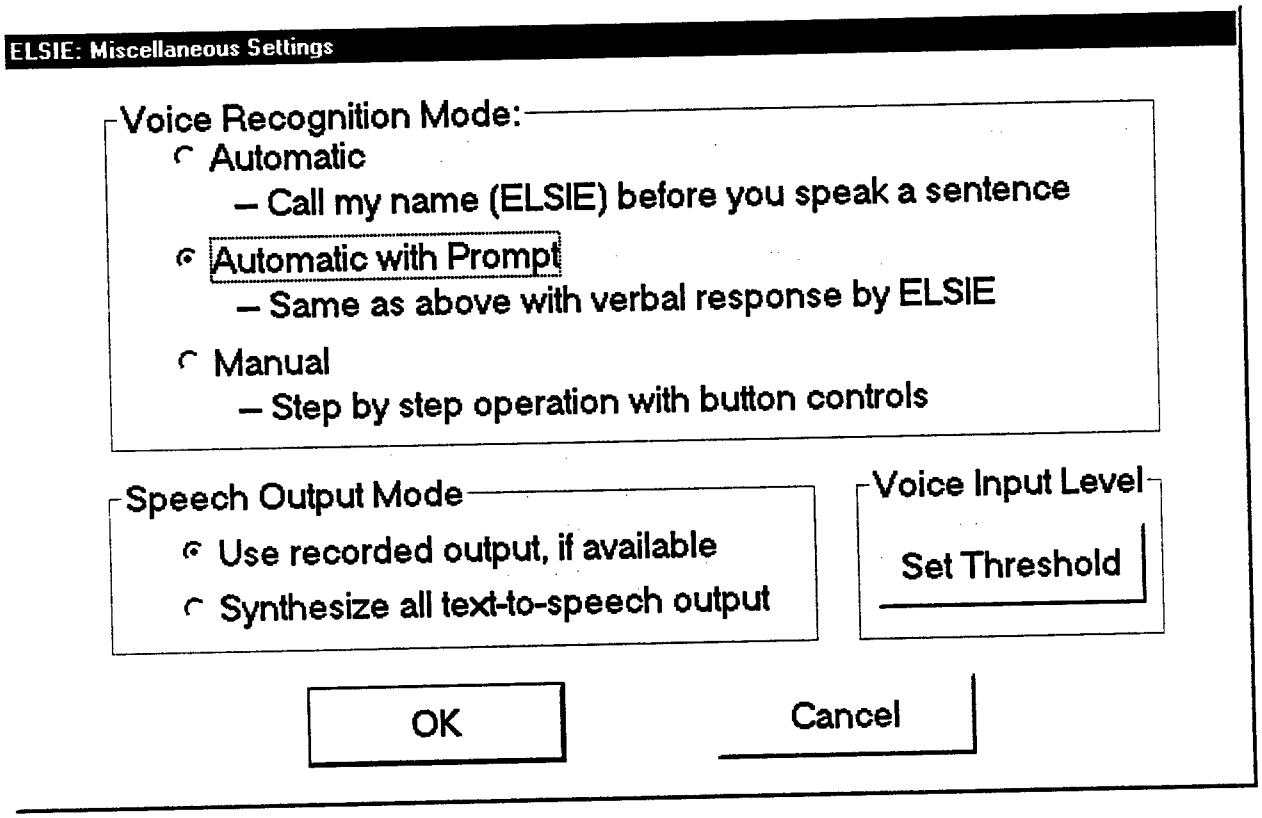
Translation:

When QRSLT/ELSIE's speech recognizer recognizes a sentence, the sentence appears in the upper box. When the sentence is translated, the translation appears in the lower box. When the program is operating in Automatic Mode, as shown, the recognition and translation occur continuously without any intervention by the user. In Manual Mode, described more fully below, the recognition and translation processes are initiated by pushing buttons.

4) Buttons for Controlling Program Operation – Along the bottom of the screen are buttons for controlling the operation of QRSLT/ELSIE:

The buttons shown above appear in Automatic Mode. The "Start" and "Stop" buttons turn the microphone on and off during Automatic Mode. If Manual Mode is selected, these buttons will instead have the captions "Record" and "Translate". In Manual Mode the microphone will not recognize a sentence by the user until the "Record" button is pushed, and it will not translate the sentence until the "Translate" button is pushed.

Pushing the "Settings" button brings up a dialog box that allows the user to select miscellaneous options for QRSLT/ELSIE:



The pushbuttons for "Voice Recognition Mode" allow the user to select between the Automatic and Manual Modes described above. ("Prompting" in Automatic Mode refers to whether or not the QRSLT/ELSIE system will answer verbally with a prompt such as "Speak now" when the user says the attention word "ELSIE" (see Section 4.1) However, in this version it is no longer necessary for the user to say "ELSIE" before speaking a sentence in Automatic Mode. The verbal response by QRSLT/ELSIE has been retained for compatibility.)

Speech Output Mode controls whether the translated sentences will be synthesized by Eloquent Technology's text-to-speech system or spoken in recorded form. Even when QRSLT/ELSIE is operating in Recorded Output mode the translation will be synthesized if a recorded translation is not available.

The "Set Threshold" button initiates a sequence in which QRSLT/ELSIE will attempt to determine the proper input sensitivity for the microphone. In previous versions of QRSLT/ELSIE it was necessary for the user to speak a test sentence in order to set the threshold. In the current version this is no longer necessary. The threshold will be set automatically and is continuously self-adjusting. Pressing the "Set Threshold" button only allows the system to change its threshold more quickly; it is a good idea to do this if ambient noise conditions suddenly change.

All of the settings above may also be changed by using verbal commands, described below.

4.2.2 "Hands-Free" Operation with Verbal Commands

Although QRSLT/ELSIE V 0.6 and subsequent versions of the system still have a "Manual Mode" in which utterances can be recognized and translated one at a time in response to GUI push-buttons, the default mode of operation for QRSLT/ELSIE is to remain in a continuous speech recognition loop and respond to verbal commands. All operations and option settings (except "Manual Mode" operations) could be initiated by verbal commands, from this version on. In V 0.6 (and subsequent versions of the system), the following categories of commands are recognized:

- **Commands for loading and unloading speech recognition contexts.**[3]

  LOAD ALL CONTEXTS
  LOAD ENGLISH-TO-SPANISH LAW ENFORCEMENT CONTEXT
  LOAD SPANISH-TO-ENGLISH LAW ENFORCEMENT CONTEXT
  UNLOAD ALL CONTEXTS
  UNLOAD ENGLISH-TO-SPANISH LAW ENFORCEMENT CONTEXT
  UNLOAD SPANISH-TO-ENGLISH LAW ENFORCEMENT CONTEXT

  Most of these context commands have one or more shortened synonyms. The words ENGLISH-TO-SPANISH may be replaced with ENGLISH and the words SPANISH-TO-ENGLISH may be replaced with SPANISH. Also, the word POLICE may be used instead of LAW ENFORCEMENT. So, for example, LOAD ENGLISH POLICE CONTEXT is the same as LOAD ENGLISH-TO-SPANISH LAW ENFORCEMENT CONTEXT.

- **Commands for setting speech recognition options.**

  SET SILENT PROMPT MODE
  SET VERBAL PROMPT MODE
  SET THRESHOLD

  The SILENT and VERBAL prompt commands turn on and off the audible response when the user says the attention-word "ELSIE" to begin recognition of an utterance. The SET THRESHOLD command prompts the user to speak a sentence which is

[3] These commands were later changed to "open" and "close" for more accurate recognition, as discussed below.

used to set the input threshold level that controls the sensitivity of the speech recognition engine.

These commands also have shortened synonyms. The first two words (e.g. SET VERBAL) are sufficient for command recognition.

- **Commands for controlling program operation.**
  PAGE UP
  PAGE DOWN
  END PROGRAM
  The PAGE commands scroll the list box that displays the sentences in the context (or contexts) currently loaded. The END PROGRAM (or QUIT PROGRAM or STOP PROGRAM) commands terminates operation of the QRSLT/ELSIE system.

- **User confirmation commands.**
  All of the commands in all categories listed above except PAGE UP and PAGE DOWN ask the user for confirmation (either YES or AFFIRMATIVE) before carrying out the command.

4.3 Changes to the Graphical User Interface for Display of Romanized Chinese

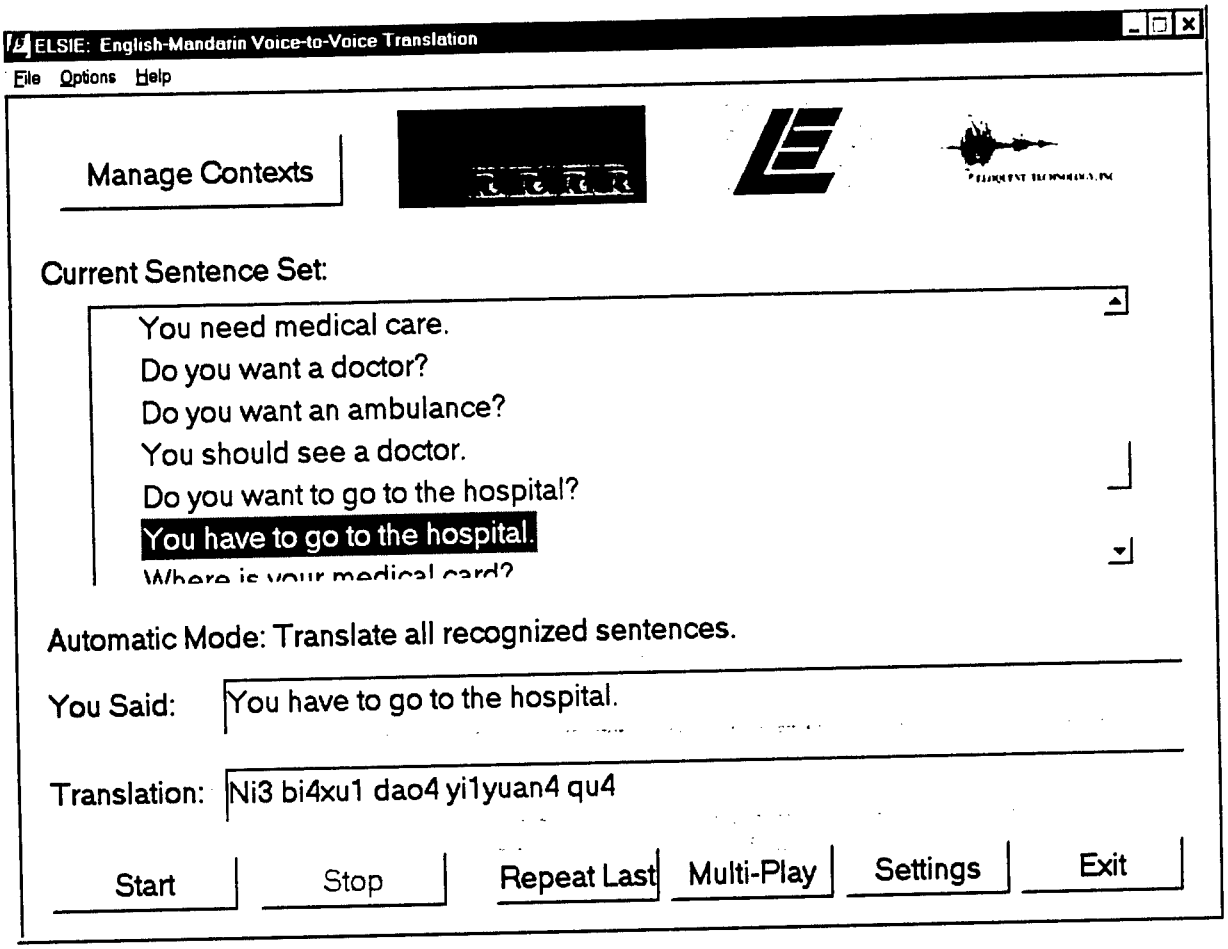
When Mandarin translation was added to QRSLT/ELSIE, it was necessary to decide how to represent the Mandarin output graphically on the screen while the recorded sentences are being spoken by the Speech Output module. There is software available for Windows 95 that would allow us to display Chinese ideograms directly on the screen, but this would be of little use to those who cannot read them (including everyone involved in the development process except LSI's Chinese linguist). However, we wanted to retain the option of displaying those characters at a later date.

There have been many systems of "Romanization" applied to Mandarin so that the language can be represented using our alphabet. Today the most widely used of these is Pinyin, which has been adopted as an official standard by the People's Republic of China.

In the official version of Pinyin, accent marks are used to identify the tone of each syllable. This representation is not convenient for representation on computer systems that display standard ASCII characters. There is a variant of Pinyin that appends a number from 1 to 4 to each syllable to identify the tone; this representation is widely used as an input method in Chinese word processing software that is designed for use with ordinary "Qwerty" keyboards. This is the method we chose for our first version of QRSLT/ELSIE that includes Mandarin.

One of the practical advantages of this method of representation is that it allows non-Chinese speaking system developers to judge the correctness of WAV file output by comparing the sound to the phonetic Pinyin representation. In one case a typing error made during revision of the Speech Output module caused the wrong WAV files to be played for several sentences; this error would have gone unnoticed by development personnel if the screen output had displayed traditional Chinese characters instead of Pinyin.

The following display shows one of the sentences from the emergency medical dialog with its Pinyin translation (see Section 4.6 for the display in Chinese characters).



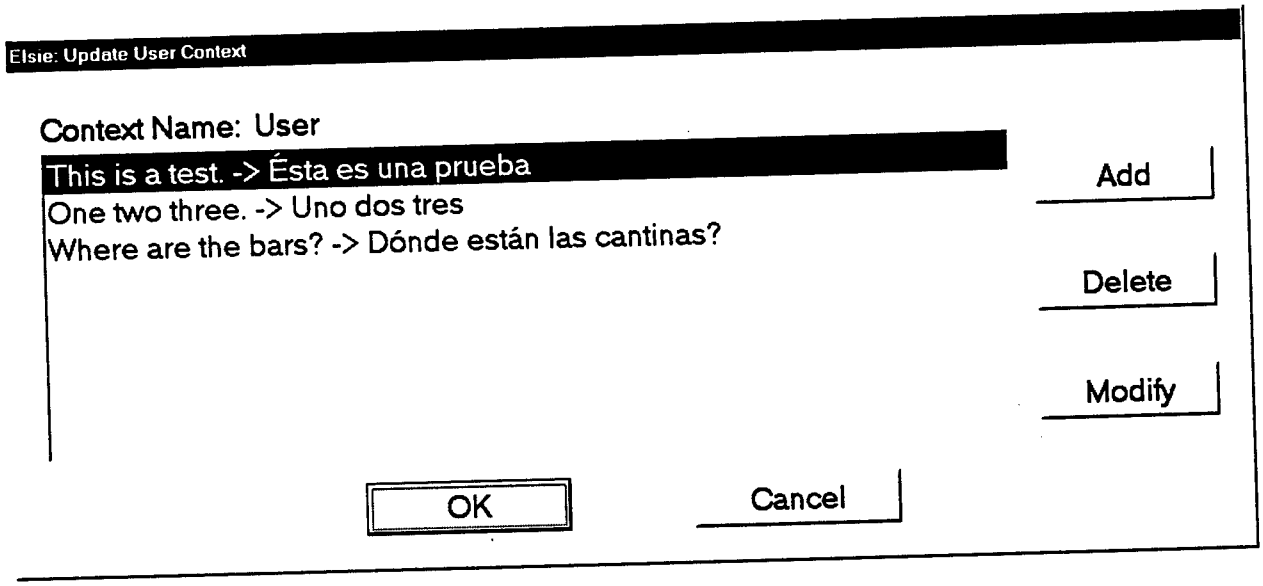
4.4 Upgrade of the User Interface: QRSLT/ELSIE V.1.2 – V.1.5

In transitioning from the 12 month to the 15 month benchmark, LSI began a rewrite of QRSLT/ELSIE's user interface. The purpose of this revision was to retain the overall appearance of the interface, but to increase functionality by switching from the simple "Dialog Box" programming paradigm to a full-featured "Document/View" interface with pull-down menus and an integrated Help system. A secondary goal of the rewrite was to incorporate Microsoft's "Data Access Objects" database engine into the program, thereby replacing QRSLT/ELSIE's ad hoc configuration files and dual-language lists of phrases and sentences with internal configuration management based on a relational database.

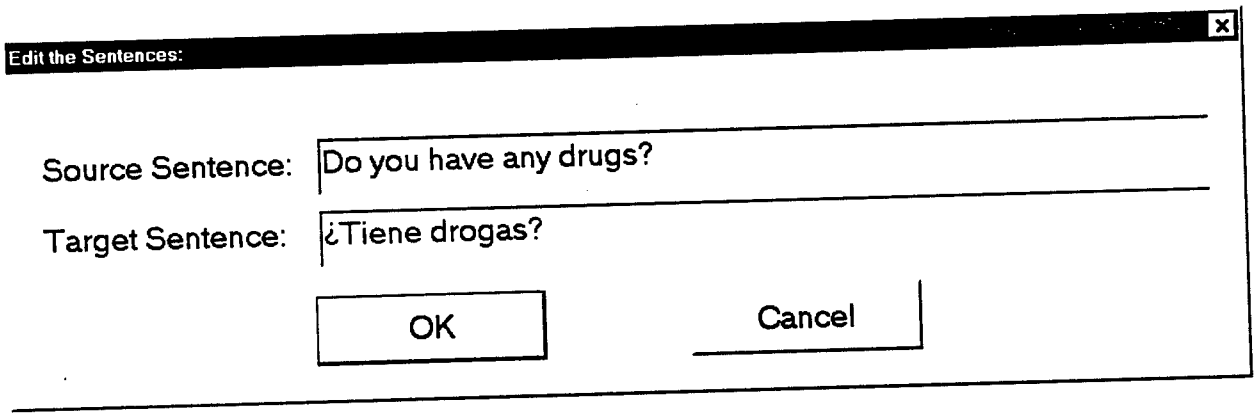
4.5 Addition of User-Defined Utterances: QRSLT/ELSIE V.1.8 – V.2.1

The upgrades discussed in the previous section resulted in increased functionality, which also allowed the addition of a capability that was frequently requested by potential users at technology expos and trade shows: the ability for users to add phrases and sentences to the system's repertoire. This capability was provided by incorporating a set of additional displays

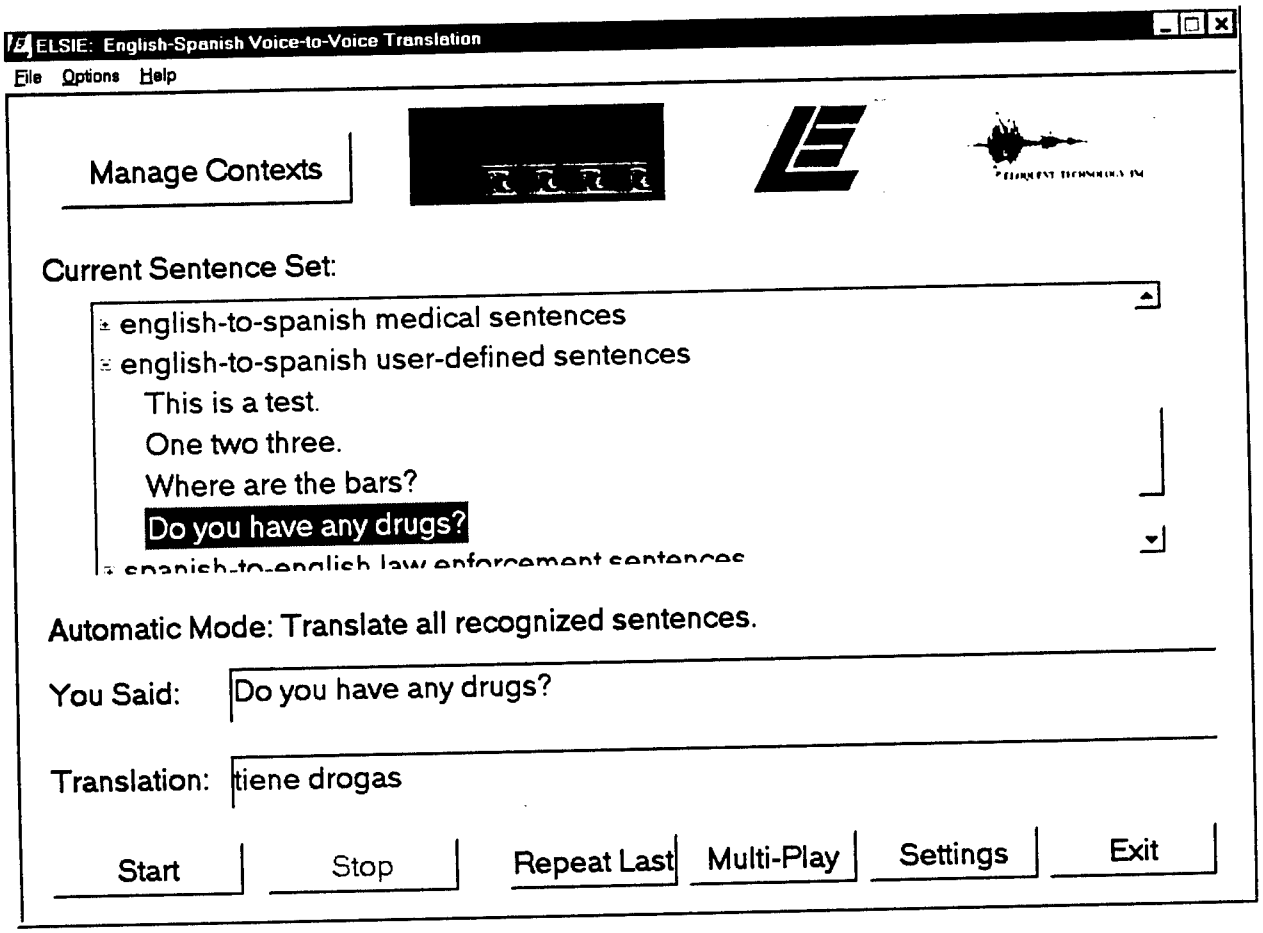
and associated functions into the user interface, beginning with the display shown below, which presents a user-expandable dialog context:



In the example, this dialog context contains three sentences. Suppose the user then wants to add the new source sentence 'Do you have any drugs?' and the translation '¿Tiene drogas?'. The user will click the *Add* button, which brings up the following display:



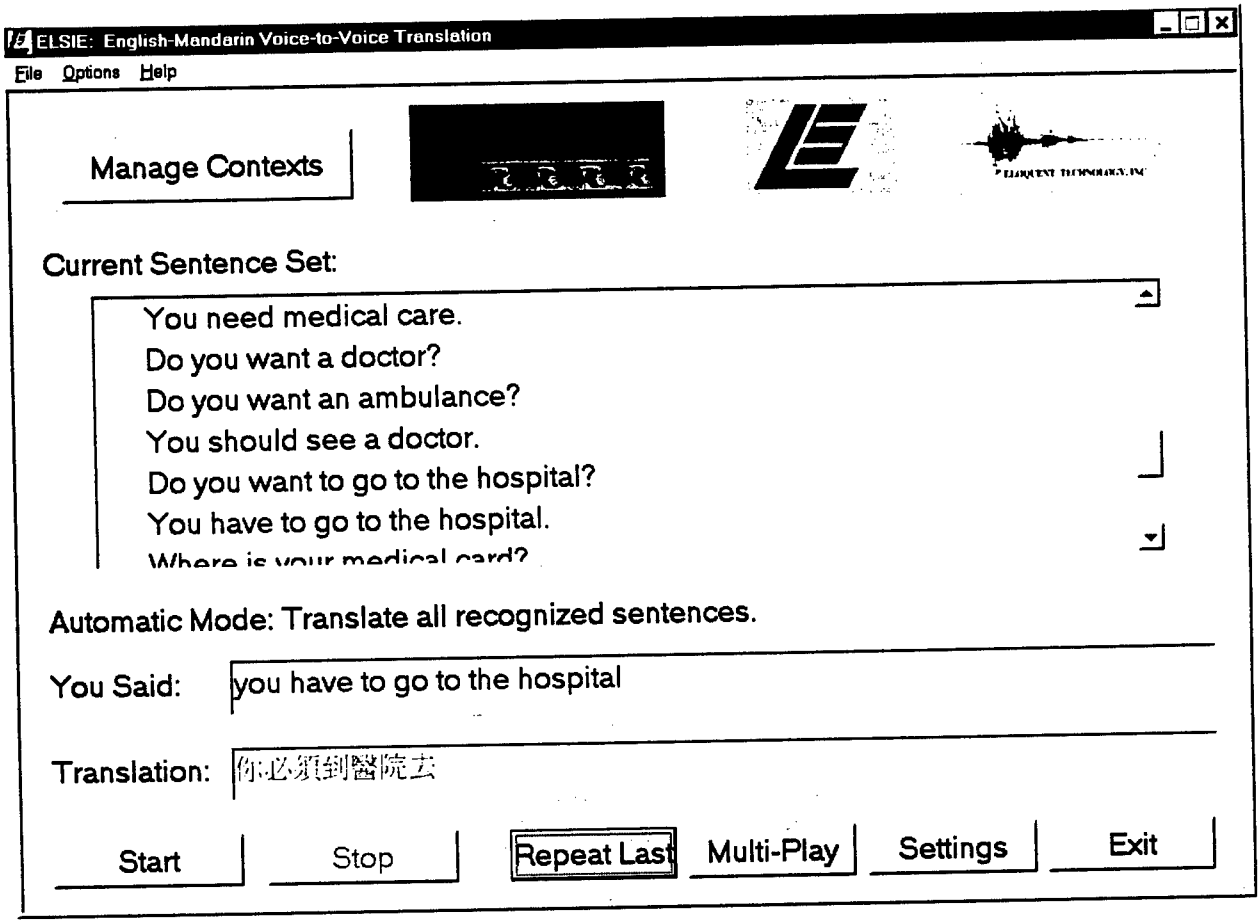
Note that both source and translation must be supplied *by the user*. This version of the system does not do free text translation, so the user must obtain and verify the accuracy of the added translations in advance, which can easily be done in a short time with the aid of a bilingual associate, or by adapting material from available bilingual manuals. By filling in the two windows with the desired sentence pair and clicking on the OK button, the new pair is automatically added to the user context, as shown below, and will be recognized, translated, and spoken just like any other sentence in the system.

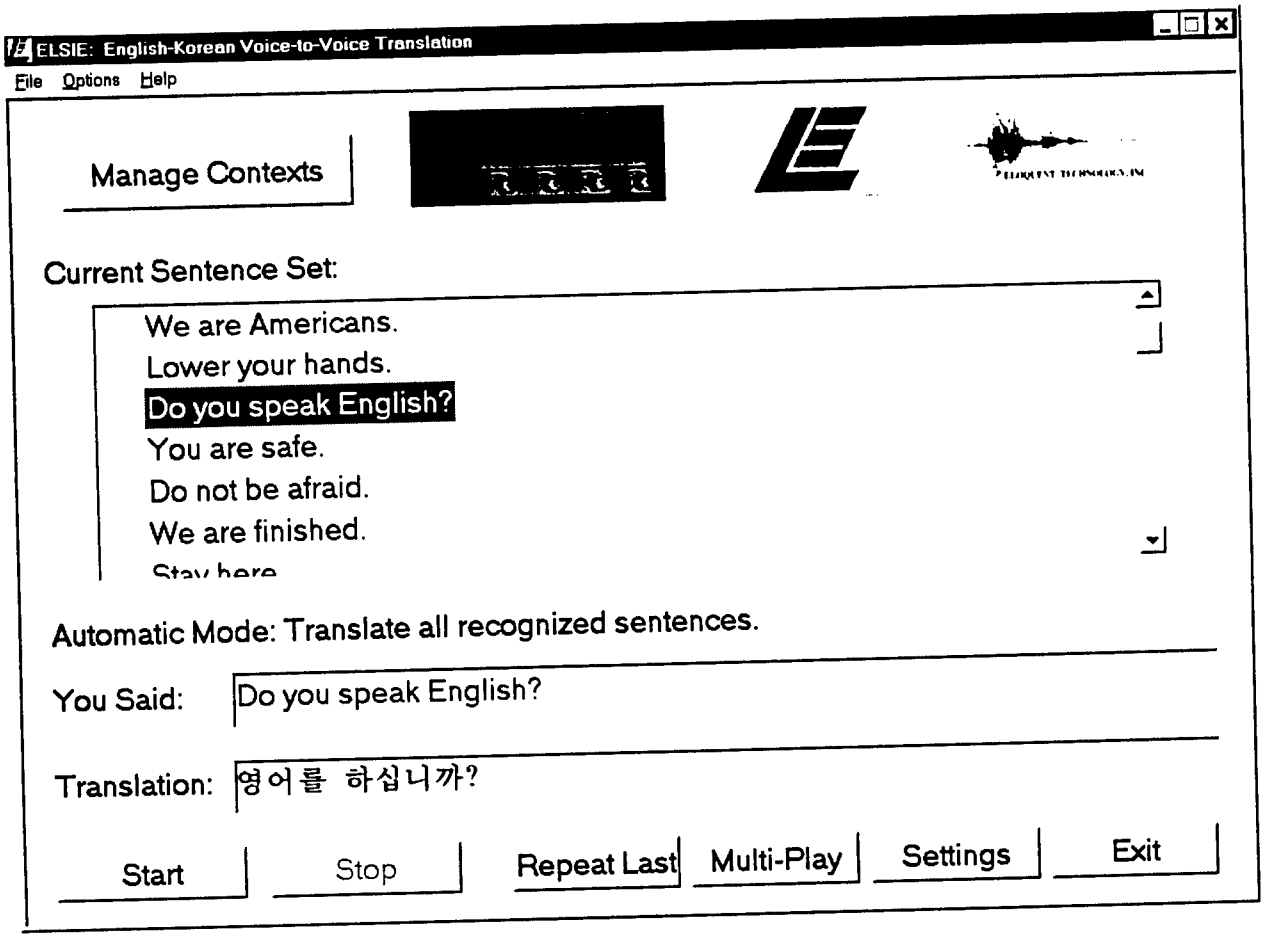


4.6 Incorporation of Asian Language Scripts into the QRSLT/ELSIE User Interface

Since the introduction of a capability for spoken translation in Mandarin Chinese in January of 1997, we conducted experiments with various software means for entry and display of Chinese characters. In addition, with the introduction of a limited Korean language translation capability, a means of entering and displaying Hangul characters was also required. For Chinese, we had been using the Pinyin romanization as a representation (see Section 4.3; however, Pinyin is not used extensively outside of mainland China, so many Chinese speakers are unable to readily understand this representation.) In the case of Hangul, there is no romanization that is generally used by Korean speakers; hence, it was necessary to acquire a software package that provided for direct entry and display of Hangul. This software, called Asian suite, also provides for entering and displaying Chinese characters.

Toward the end of the project period, we developed an integrated capability for representing either character set on a display, using the HTML display capability of Microsoft Explorer to display the characters input via the Asian Suite software. For Korean, the input codes are also converted to a romanized representation which is useful for non-Korean speaking system developers. In the case of Chinese, both Pinyin and the ideographic representation are input. A switch in the "Options" menu allows users to select either the romanized or character representations for both Chinese and Korean. Examples of Chinese and Korean displays are shown below, and in Appendixes B and D.


ELSIE: English-Mandarin Voice-to-Voice Translation
File Options Help
Manage Contexts
Current Sentence Set:
You need medical care.
Do you want a doctor?
Do you want an ambulance?
You should see a doctor.
Do you want to go to the hospital?
You have to go to the hospital.
Automatic Mode: Translate all recognized sentences.
You Said:
you have to go to the hospital
Translation:
你必須到醫院去
Start
Stop
Repeat Last
Multi-Play
Settings
Exit


ELSIE: English-Korean Voice-to-Voice Translation
File Options Help
Manage Contexts
Current Sentence Set:
We are Americans.
Lower your hands.
Do you speak English?
You are safe.
Do not be afraid.
We are finished.
Automatic Mode: Translate all recognized sentences.
You Said:
Do you speak English?
Translation:
영어를 하십니까?
Start
Stop
Repeat Last
Multi-Play
Settings
Exit

5 MARKET ANALYSIS AND COMMERCIALIZATION ACTIVITIES

5.1 General

As soon as a QRSLT/ELSIE prototype was available, LSI and Rome Laboratory personnel began to show it to military, law enforcement, state and local government, and other contacts in an effort to gather information about the reactions and requirements of potential users. Many of these demonstrations took place at technical and trade shows, and at other venues arranged through contacts from these shows or through other channels.

Many of these contacts with potential purchasers or users of voice translation have a predictable and now familiar pattern. At first, there is surprise that such a product is even possible. This is often followed by an overestimation of the capabilities of machine translation and speech recognition. Visitors to an exhibit at a technology expo may attempt to engage the system in free conversation, for example. When visitors stay and listen to our presentation of the system and try it themselves with a clearer understanding of its purpose and limitations, typically there is lively interest in both the immediate and long term commercial possibilities of the system. This is often found even in visitors who themselves are not potential users. We have gained a number of interesting and intelligent suggestions about possible applications from such visitors.

In particular, as mentioned elsewhere in this report, many of these potential users made invaluable suggestions about additional features and functionality that they would consider important for a given application. To the extent possible, within the available resources of this cooperative agreement, we have developed such functions and incorporated them into the system, as described in the appropriate sections of this report.

In the following discussion, we summarize the events and venues in which LSI demonstrated the system as part of the market analysis and commercialization efforts which are specified in this cooperative agreement. The technology expos and demonstrations are divided into three sections: Sections 5.2 and 5.3 address the military and law enforcement applications which were the specific topics at which this project is directed; Section 5.4 lists conferences and events which fall outside these primary topics of interest. In addition to the conferences and technology fairs, demonstrations and visits with local law enforcement, military, health, and social service agencies are briefly described below under the appropriate section heading.

5.2 Technology Demonstrations and Exhibits: Military Applications

US Marine Corps – Camp Pendleton

On January 7, 1997, LSI staff members demonstrated the QRSLT/ELSIE and the MAVT prototype system to USMC Special Forces elements at Camp Pendleton, CA. Attendees included Special Forces personnel from intelligence, operations, and logistics, all of whom expressed interest in spoken translation for their activities.

Special Operations/Low Intensity Conflict (SOLIC) Conference: 1997-8

On February 11-13, 1997, the Rome Laboratory Project Manager, Sharon Walter, and Christine Montgomery of LSI exhibited QRSLT/ELSIE at the SO/LIC conference in Washington, D.C. to AFSOC and SOCOM officers, as well as other attendees of the conference. The version of the system demonstrated included Spanish and Mandarin Chinese.

On February 17-19, 1998, Christine Montgomery of LSI and Sharon Walter of Rome Laboratory exhibited the QRSLT/ELSIE system at the SOLIC conference in Crystal City. Members of the Special Operations Command showed particular interest in the system, and have requested further demos and briefings for their respective units in the command.

SSCOM Warrior Advance Planning Briefing and Exhibition

On September 2-4, 1997, in Cape Cod, MA, LSI exhibited the QRSLT/ELSIE system at this Army conference, which focused on "Equipping, Protecting and Sustaining the Warrior". The spoken translation system aroused a great deal of interest among the approximately 300 conference attendees. In particular, a general, also a medical doctor, in the Air Force Reserve, expressed interest in utilization of the system for medical interviews, and members of the Military Police saw uses paralleling our law enforcement scripts.

Coalition Warfare Task Force Meeting

On March 4,1998, Christine Montgomery of LSI and Dan Benincasa of Rome Laboratory presented a briefing and demonstrated the QRSLT/ELSIE system before a meeting of the Coalition Warfare Task Force of the Defense Science Board at the Army War College in Carlisle, PA. The objective of the meeting was to show feasibility of multilingual technology and the need for further investment in this area. The invitation to participate in the meeting came from the Army Research Laboratory.

Army TOC Technology Symposium - Huntsville

On April 14 - 15, 1998, LSI personnel showed the upgraded version of QRSLT/ELSIE at this conference, which showcased advanced technology for Army Tactical Operations Centers.

Global Patriot Exercise, Shaw AFB, SC

During this final period of the project, LSI extended and enhanced the Korean language capabilities of the QRSLT/ELSIE system for use during the Global Patriot exercise. (Details of these enhancements are presented in Appendix D.) On July 14 - 17, Sharon Walter of Rome Research Site, AFRL, and Christine Montgomery of LSI demonstrated the QRSLT/ELSIE system in the AF operations center for the exercise at Shaw AFB, one of three military centers participating in this exercise. A number of military personnel experimented with the system, focusing mainly on the Korean and Spanish language capabilities. In general, they were impressed with the system's functionality, and suggested a number of military and law enforcement applications in which the QRSLT could be used.

USAJFKSWCS, PsyOp Div, DOTD, Ft. Bragg, NC

On August 12, Christine Montgomery of LSI demonstrated the QRSLT system to staff members of the PsyOp Division at Fort Bragg, following up on an invitation from the former chief of that group, Col. Romeo Morissey. Members of the group were interested in obtaining a copy of the system for test and evaluation in the near future, commenting on its potential utility for both operations and language training.

5.3 Technology Demonstrations and Exhibits: Law Enforcement

Technology for Community Policing, San Diego: September, 1996

On September 9-10, LSI and Rome Laboratory accepted an invitation from the law enforcement organizations represented in the Border Research Center in San Diego to participate in a conference entitled "Technology for Community Policing". Sharon Walter, RL/IRAA, demonstrated the portable MAVT ADM system installed on a Sun Voyager, while LSI staff members showed the alternate version of QRSLT/ELSIE with speech recognition via a beta version of the IBM Voice Type Application Factory (VTAF) -- a successor of the IBM Continuous Speech Series (ICSS) (described in Section 6.2). Both systems performed well in spite of a high level of ambient noise emanating from an adjacent booth showing a training video featuring gunfire and other high volume special effects.

Law Enforcement and Corrections Technology Symposium

From May 19 - 22, 1997, Christine Montgomery of LSI demonstrated QRSLT/ELSIE at this law enforcement technology exposition in Orlando, FL. Interested potential clients included a number of police departments, sheriff's departments, and highway patrol departments, mainly from the eastern U.S. Information was provided to Dr.Montgomery on the NIJ technology testbed program, as well as on the ALERT vehicle project.

The ALERT (Advanced Law Enforcement Response Technology) Project is sponsored by USDOT, the Texas Department of Transportation, and the Texas Transportation Institute. The objective is to introduce integrated information technology within the patrol car, facilitating the collection of information, completion of required paperwork, and access to criminal justice data bases. The current version uses a ruggedized PC located in the vehicle's trunk, and inside the car, a touch-screen display and a handheld remote unit, which contains the standard forms to be completed. Several of the officers who associated with the ALERT project suggested the integration of QRSLT/ELSIE into the vehicle's PC system.

Meeting with LA County Sheriff's Department Century Station Personnel

On June 25, 1997, LSI staff visited one of the newer facilities of the LACSD to observe operations, interview LACSD personnel, and collect various arrest and booking forms to assist in defining requirements for law enforcement use of QRSLT/ELSIE.

Installation of QRSLT/ELSIE at the Fresno County Sheriff's Department

A major accomplishment during this reporting period was installing a QRSLT evaluation system for the Fresno jail booking application at the Fresno County Sheriff's Department custodial facility. This system was installed on a PC notebook computer on August 6, 1997, and has been in experimental use for booking Spanish-speaking prisoners into the jail since installation. Fresno County SD personnel were contacted at the LA Government, Business, and Education Technology Expo in October, 1996, and LSI has followed up on these contacts.

Prior to the installation visit, a videotape demonstrating the functionality of the year 1 QRSLT/ELSIE system was prepared and copies were shipped to the LA and Fresno County Sheriff's Departments.

The Fresno County Sheriff's Department responded immediately, stated that the QRSLT system development had progressed much more rapidly than they had thought possible, and invited us to visit Fresno for the semimonthly meeting of the command staff associated with the custodial operations at the jail. LSI staff members spent August 5th and 6th observing booking operations at the jail, discussing procedures with the Sheriff's Department officers and staff, and demonstrating the system. The QRSLT system was then installed on a notebook computer, and, after a brief hands-on training session, a sergeant from the Fresno SD Jail Division assumed the role of user/operator and trainer for other personnel involved in the booking operation. In talking with her a few days later, we found that the system was being experimentally tested in the booking operation.

Visit by LA County Sheriff's Department Officers

On November 10, 1997, the new technology representatives for the LACSD visited LSI to view a demonstration of the current system, and to discuss introduction of QRSLT evaluation system into the department. The LACSD is beginning a "Station Automation" initiative, and the officers were particularly interested in the potential of the QRSLT technology for this development.

International Association of Chiefs of Police (IACP): 1997-8

On October 25-28, 1997, in Orlando, LSI exhibited the QRSLT/ELSIE system at this large law enforcement conference, which is attended by representatives of law enforcement agencies in the US and abroad. In terms of conference size and attendance at the exhibits, this conference exceeded all previous records for exposure of the QRSLT system. Contacts made at the conference were followed up, and many of the persons who saw a demonstration there contacted us directly for more information.

In 1998, LSI personnel attended this conference and technology expo in Salt Lake City. Interest was expressed by the National Park Service Rangers, Border Patrol and Customs officers, and several representatives of foreign police services.

International Land Transportation Security Conference - Atlanta

On April 9, 1998, Christine Montgomery of LSI showed the QRSLT/ELSIE system in the NLECTC booth. Officers from a number of law enforcement and transportation security agencies throughout the US attended this conference. Discussions were held with the Department of Transportation office and the system development organization for the ALERT patrol car. These agencies were interested in adding the spoken translation capability to the information technology being showcased in the ALERT car.

Arizona Law Enforcement Expo 98 – Phoenix

On April 30, LSI personnel exhibited the QRSLT/ELSIE system at this technology expo, which is sponsored by the Arizona Association of Chiefs of Police (AACOP). In addition to the obvious relevance of Spanish dialogs, attendees at this conference named Japanese as the next most important language, due to the large number of tourists visiting Grand Canyon and other tourist sites in Arizona.

National Law Enforcement Criminal Justice Expo - Los Angeles

On May 6 - 7, 1998, LSI staff members demonstrated the QRSLT/ELSIE system at this conference, which was attended by personnel from a range of local law enforcement agencies.

Border Law Enforcement Technology Expo, Albuquerque, NM

On May 19, 1998,Christine Montgomery of LSI exhibited the QRSLT system at this conference at the invitation of the Border Research Center in San Diego. The system was demonstrated to Dr. David Boyd, NIJ Director, in addition to law enforcement and military personnel attending the meeting.

L A County Sheriff's Department

On May 26, 1998, Christine Montgomery gave a lecture and demonstration of the QRSLT/ELSIE system at a management meeting of the L A County Sheriff's Department. On May 28, LSI staff members visited the LASD Century Jail to install the upgraded version of the QRSLT system in the booking area of the jail facility for hands-on experimentation and evaluation.

5.4 Participation in Technology Expositions and Trade Shows: Other Applications

Los Angeles Government, Business and Education Technology Expo: 1996-7

From October 16 through October 18, 1996, LSI exhibited QRSLT/ELSIE and MAVT ADM at the Greater Southern California Government, Business, and Education Tech Expo 96 held at the Los Angeles Convention Center. This tech expo attracts large numbers of representatives from state and local government agencies, many of whom identified potential applications for QRSLT/ELSIE within the operations of their agencies. In particular, personnel from the Fresno County Sheriff's Department described a jail booking application for which they would like to

purchase QRSLT/ELSIE when available, and representatives of an agency which provides emergency response for child abuse cases in three Southern California counties were also interested in working with us to develop scenarios for their application. Discussions by telephone and email were continued with both these agencies (see below).

On September 30 - October 2, 1998, LSI again exhibited the QRSLT/ELSIE system at this conference. Several interesting contacts were made at the conference, in particular, teachers concerned about communicating with parents of students, and an LA County government social services representative charged with supervision of nutrition for the patients at local nursing homes. Her job involve contracting with local restauranteurs for preparation of food and education/monitoring of nutrition requirements for the nursing homes. Her primary language requirements were for Asian languages, and she was extremely interested in acquiring a spoken translation system that would assist her in communicating in those environments.

SpeechTek, New York: 1996-7

On October 22 and 23, 1996, LSI exhibited ELSIE and the MAVT ADM Voyager system at the SpeechTek technology exposition held at the New York Hilton. Applications identified by prospective users included public transportation, emergency medical services, and tourist information. Contacts were also made with hardware developers of handheld PCs and hardware vendors interested in value-added features incorporating spoken translation.

On September 30 and October 1 in New York City, LSI exhibited the QRSLT/ELSIE system at this conference, which is devoted to leading edge speech technology. Eloquent and Entropic were also exhibitors for text-to-speech and speech recognition, respectively, at SpeechTek. The QRSLT/ELSIE system received some media attention, including CBS radio and television. A number of contacts were made with companies interested in bundling QRSLT as value-added software with their systems.

AVIOS 97

On September 9-11, 1997, in San Jose, LSI exhibited the QRSLT/ELSIE system at the AVIOS 97 conference, another leading edge speech technology venue. Eloquent and Entropic were also exhibitors for text-to-speech and speech recognition, respectively.

Technology 2006

From October 29 through October 31, 1997, LSI exhibited QRSLT/ELSIE and the MAVT ADM Voyager system at the Technology 2006 exposition in Anaheim, California. Applications identified at this technology expo included educational and business activities.

Accelerating Technology 97-8: Riverside County, CA

On March 20, 1997, and March 19, 1998, LSI staff members demonstrated QRSLT/ELSIE at this technology exposition in Riverside CA. The expo was organized by the county Department of Information Services, and was attended by over 2,000 persons, primarily state, county, and

city government personnel. A number of useful contacts for QRSLT commercialization activities were made, including a police chief who controls an area where 95% of the residents are Spanish speakers, a teacher who deals with first grade classes containing non-English speaking children whose native languages are Spanish and a number of Asian languages, and personnel from a Riverside County mental health facility where most of the patients are Asian language speakers. Social work managers expressed an interest in learning more about the system and in assisting with necessary customization.

Meeting with Riverside County Social Services Personnel

On April 10, 1997, members of LSI's staff visited personnel of the Riverside County Social Services Department to follow up their interest in using QRSLT/ELSIE for visitations in connection with emergency response to child abuse situations. Although they had collected some taped interviews for us, they were in the midst of a large state-wide data base creation effort, which would hinder any other development work for some time.

Technology and Persons with Disabilities - Los Angeles (1997, 1998)

This is among the largest and best-attended conferences in assistive technology. It is sponsored annually by the Center on Disabilities of California State University at Northridge. LSI personnel demonstrated QRSLT/ELSIE during the four days of this conference (March 18 – 22, 1997). We also attended in March, 1998, again gathering a number of ideas and individual contacts.

Among the kinds of applications and configurations suggested at this conference were dialogs oriented to communication with care-givers in home settings and in institutions, as well as the development of a hand-held module.

IEEE Dual-Use Technologies and Applications Conference

On May 13, 1997, Christine Montgomery presented a paper and a demonstration of QRSLT/ELSIE for a law enforcement panel chaired by Sharon Walter, the QRSLT program manager for Rome Laboratory, at the Dual-Use Technologies and Applications Conference held in Syracuse, NY.

Local and Regional Healthcare Organizations

In early June, 1997, Dr. Christine Montgomery of LSI demonstrated QRSLT/ELSIE for a physician member of the board of Catholic Healthcare West (CHW), a consortium of health service organizations operating in the southwest.

On June 10, Christine Montgomery and Dr. John Fought followed up on the earlier meeting in a visit with Elinor Ramirez, who is in charge of nursing operations at St. Vincent's Hospital in Los Angeles, a member hospital of the CHW organization.

In both meetings, interest was expressed by the health service personnel in utilizing such a system for some interview situations, but their language requirements tend to be for translation of languages other than Spanish and Chinese (e.g., Korean).

MT Summit VI - San Diego

On October 30, 1997, Christine Montgomery participated in the Pioneers Panel at this conference on "Machine Translation Past, Present, and Future", which commemorated 50 years of MT. She discussed the QRSLT project as an indicator of future technology directions -- a concept which was reinforced in several other presentations at the conference.

Meeting of the American Immigration Lawyers' Association

On November 21, 1997, LSI staff members exhibited the QRSLT/ELSIE system at a meeting of the California chapter of this organization. Several of the attorneys who saw the system demonstration expressed interest in participating in the development of relevant immigration law dialogs in return for discounts on the resulting product.

La Opinion's 'Technology for the Red Familiar (Family Network)

This technology fair is one of many community outreach programs organized by LA's prinicipal Spanish language daily newspaper, *La Opinion*. LSI's participation in this event resulted from contacts with members of La Opinion's staff at the LA Technology Expo in October. John and Carmen Fought exhibited QRSLT/ELSIE, which was one of the most-viewed demonstrations at the fair, and received prominent coverage in *La Opinion's* report on the event. A number of worthwhile suggestions, many involving possible applications for retail sales chains, were received from visitors.

Government Technology Conference West – Sacramento

On May 13 - 15, 1998, staff members of LSI exhibited the QRSLT/ELSIE system at GTC West -- a large annual conference held to acquaint state, county, and city government personnel in the western US with new and emerging technology. Attendance at the conference was estimated at 22,000. Government personnel who interacted with the system and expressed interest in utilizing it came from a range of agencies, including law enforcement, social services, disaster relief, motor vehicles, and education.

6 SPEECH PROCESSING COMPONENTS: SPEECH RECOGNITION

In the course of QRSLT/ELSIE system development, several different speech recognizers from three companies -- IBM, Dragon, and Entropic -- were utilized in various versions of the system. As noted previously, the operating concept established at the first technical meeting was to utilize commercial recognizers available for PC-Windows until Entropic had ported their Unix-based HTK system to the PC-Windows environment. This section describes experimentation with all of these recognizers, integration into the QRSLT/ELSIE system, speed and accuracy, and other relevant characteristics within the spoken translation application. With the exception of the work performed by Entropic, as described in Section 9, all speech recognition development, integration, and testing was performed by LSI.

6.1 Initial QRSLT v.03 /QRSLT/ELSIE 1 with Dragon Dictate

The QRSLT 3-month milestone was achieved on August 15, 1996, by configuring an initial system with basic functionality for the three processing components of speech recognition, language translation, and speech generation for one-way translation from English to Spanish. The speech recognizer used was Dragon Dictate, a speaker-dependent, isolated word recognizer developed by Dragon Systems. The test corpus was a set of 30 sentences drawn from the military command and control cards and from the initial law enforcement materials provided by the LA County Sheriff's Department (see Section 2). LSI used a direct translation strategy for the initial test, as described in Section 3. Spanish translations of the English sentences were generated as wave forms prerecorded by Eloquent. Entropic performed the integration and testing of this initial system.

Dragon Dictate (DD) had been selected by Entropic as the commercial recognizer to be used in the initial system because it could run under both the Windows-95 and NT operating systems. The fact that it was speaker-dependent was not that significant for the initial system, since only one-way translation was to be demonstrated. In order to overcome the isolated word limitation of Dragon Dictate and simulate continuous speech, the English input phrases and sentences were treated as single words. Although DD's interface for spoken commands and vocabulary additions was easy to use, constructing a path through the system was complicated by the fact that the process for dictation correction could not be bypassed in testing, and that DD retained control of the audio card, making it necessary to shut down DD to generate output speech using Eloquent's Software. In addition, DD's macro and script language executed very slowly in command mode, making the use of spoken commands rather more of a necessity than a virtue.

6.2 Alternate QRSLT/ELSIE with IBM VTAF

Because of the problems encountered in using Dragon Dictate, LSI began experimental development of an alternate system using a beta version of IBM's Voice Type Application Factory (VTAF – formerly called ICSS: IBM's Continuous Speech System) for recognition. As in the initial system with the DD recognizer, a direct translation methodology for conversion of all sentences from English to Spanish was utilized, and Spanish output was generated via prerecorded wave files. VTAF allowed the use of an "attention context": a one-word context which signals to the system that it should begin listening for spoken input. The selected attention

word was "ELSIE". When addressed by users, QRSLT/ELSIE gave one of a number of prerecorded responses, indicating the system was ready to accept input.

6.2.1 Description of the API

IBM's VTAF consists of a set of Dynamic Link Libraries (DLLs) containing C language functions that can be used to perform speaker-independent speech recognition and .WAV sound output on a 32-bit MS-Windows platform. The VTAF functions access the PC's sound card through the standard Windows multimedia interface, so no special hardware is required. Along with the DLLs, the package includes a set of programs used to perform utility functions such as compiling words and grammar rules into a "context file" (or speech grammar), and setting the sound threshold for a recognition session.

6.2.2 Development of the Recognition Component

The initial step in this first version of QRSLT/ELSIE was to compile the sample military/law enforcement dialog corpus (see Section 2) into a .CTX "context file" (speech grammar) that could be used by VTAF routines. This was accomplished by creating a .BNF text file specifying the allowable grammatical constructs and vocabulary, and then processing it with the context compiler. For this first version, we limited the system to the recognition of complete sentences. The first part of the .BNF file was as follows:

```
<utterance>   ::= <sentence1>
                | <sentence2>
                | <sentence3> ...
```

The second part of the .BNF file defined the content of the sentences:

```
<sentence1>  ::= We are Americans .
<sentence2>  ::= Lower your hands .
<sentence3>  ::= Do you speak English . ...
```

A dictionary file supplied with the VTAF package contained all the words needed for our sample sentences, although VTAF does allow the user to create new dictionaries by supplying phonetic definitions for new words.

6.2.3 Using the Recognition Context in the QRSLT/ELSIE Program

After the contexts were created, we incorporated the sample sentences, their Spanish translations, and the names of the .WAV files containing the spoken output into a .DAT text file as follows:

We are Americans.
Somos americanos.
DELTA1.WAV
Lower your hands.
Baje las manos.
DELTA2.WAV

This file is read by the QRSLT/ELSIE program when the user selects the .CTX file. Translation and spoken output take place based on a direct lookup of utterances and their translations.

6.2.4 Initializing and Using the VTAF Functions

From within the QRSLT/ELSIE program, the VTAF system was initialized by calling the ICSSStart() StartConversation() functions. This caused a separate application window (ICSWINMM.EXE) to be spawned in the background. The contexts were then loaded using the ICSSLoadContext() function. The speech recognition is performed with a pair of functions, ICSSListen() and ICSSGetSpokenWords(). The latter takes a pointer to an ICSS_RETURN_ WORDS structure as its argument; if the function returns successfully, this structure contains the sentence recognized by VTAF.

Spoken output was produced by calling the ICSSPlayback() function and passing the name of a .WAV file to it.

6.2.5 Programming Considerations: Ownership and Scheduling

An important factor to consider in coordinating speech recognition and sound output on a PC is that typically only a single subsystem can control a PC sound card at a time. In this case, we used the IBM VTAF system for both input and output, so there was no need to perform time-consuming shutdown and reinitialization of the sound hardware when switching from input to output.

Even though input and output were performed by the same subsystem, the scheduling of these functions required special care. During the execution of the ICSSGetSpokenWords() function, the executing process no longer responded to input or control messages from the operating system. To keep the system responsive and allow updating of the user interface window, it was necessary to spawn off "worker threads" to perform the speech recognition task. At the end of a recognition cycle, the worker thread sends a user-defined window message back to the message loop of the main interface thread to signal it to perform the translation.

The same scheduling consideration came into play with sound output. The ICSSPlayback() function was also called from its own worker thread. Even with separate execution threads, sound input and sound output could not take place simultaneously. An input or output call would simply "block" until the previous function completed. For this reason, it became clear during testing of the user interface that we needed to provide the user with an easy-to-see visual prompt during the time that the system was listening for input.

6.2.6 Performance Analysis of VTAF Speech Recognition

The VTAF system did well in its recognition of sentences, but was less reliable in its recognition of the attention word to begin an utterance. Many users found it necessary to say the attention word several times before the system recognized it. Once a normal recognition sequence was initiated, reliability of the sentence recognition exceeded 90%.

The response time was excellent. On a 133 MHz Pentium laptop, the recognition appeared to the user to be immediate, and gave a smooth transition from spoken input to translated spoken output. Of course, the translation time was minimal, because it was achieved via a direct translation strategy of table lookup in this initial version of the QRSLT/ELSIE system. More

complex versions of the translation strategy described below inevitably introduced a longer delay between input and output; minimizing this delay during parsing and translation proved to be a challenging exercise in program optimization, which continued throughout the project.

6.2.7 Advantages and Disadvantages of VTAF

The primary advantage of VTAF, that it could be used as a simple, "black-box" approach to speech recognition, was also its main disadvantage. The programmer who enjoyed VTAF's ease of use sacrificed the flexibility that would be available with an API that allowed finer control of low-level operations. Other than the listening and playback functions described earlier, the VTAF API had no facility for adjusting performance except a function that allowed the programmer to set a few internal numeric parameters. The effect of changing these parameters was unclear and not well documented in the beta version of the system that was made available to us.

Another disadvantage of the "black-box" nature of VTAF was that it could not receive feedback from the translation portion of the program that would allow it to improve its recognition. As QRSLT/ELSIE progressed through subsequent versions of the system, one of the main goals was to create a tighter coupling between the recognition and translation processes to achieve the design objectives of speed and accuracy. Unfortunately, this goal was not to be realized in the course of the project, since neither of the IBM ASRs used made this possible, and only an alpha version of the PC Windows-based HTK recognizer was delivered to LSI by Entropic in the course of the project (Sections 6.6-7.).

6.3 Evaluation of the 3 Month Benchmark Systems

When the performance of the two versions was compared at a meeting of the consortium members, it was decided that the system based on IBM's ICSS better fulfilled the original design goals of continuous, speaker-independent speech recognition and interactive use. Thus, the alternate version of QRSLT/ELSIE became the basis for developing QRSLT/ELSIE Version 0.6.

The consortium members agreed on the following goals for features to be incorporated into Version 0.6 of QRSLT/ELSIE:

- Hands-Free Operation -- The target hardware platform for the final version of QRSLT/ELSIE was to be a hand-held or belt-mounted portable computer. Assuming that the portable computer would not have a standard keyboard or mouse-driven user interface, we believed that it was important for the user to be able to control the program entirely through verbal commands.
- Simultaneous Multiple Language Recognition – The alternate version of QRSLT/ELSIE shown at the consortium meeting already had the capability to recognize and translate Spanish as well as English sentences chosen from the military/law enforcement corpus, but only one language could be recognized at a time; switching languages required unloading one recognition context and loading another. It was decided that it would be desirable for the system to be sensitive to both languages at the same time.

- New Medical Context – The consortium agreed that the current set of military/law enforcement-related sentences would be augmented with a set of sentences based on an emergency medical scenario.
- Possible Inclusion of Spanish-Language Speech-Synthesis – Version 0.3 of QRSLT/ELSIE used only recorded sentences as Spanish-language output. It was agreed that Version 0.6 would continue to use recorded output both for the existing military/law enforcement sentences and the new medical sentences. However, Eloquent Technology offered to provide an alpha version of their Spanish Text-To-Speech system if it became available before Version 0.6 is finalized (which was provided, and included in the V 0.6 system).

6.4 QRSLT/ELSIE Version 0.6

With these goals in mind, LSI began redesign and development of the alternate QRSLT/ELSIE system to produce the enhanced functionality of V 0.6, most features of which were retained in subsequent versions of the system. Design and implementation of the goal of hands-free operation is discussed above under the user interface (Section 4).

6.4.1 Simultaneous Multiple Language Recognition

As noted previously, Version 0.3 of QRSLT/ELSIE could recognize and translate sentences from a military/law enforcement corpus in either English or Spanish. However, only one recognition context could be used at a time, so English and Spanish speech recognition had to take place in separate sessions.

For Version 0.6, the speech recognition module was redesigned to allow multiple contexts and multiple languages to be used simultaneously. The default for Version 0.6 (and subsequent versions of the QRSLT/ELSIE) is to load both Spanish and English contexts and activate them when the program runs. This gives QRSLT/ELSIE the ability to recognize both English and Spanish at the same time. The user may choose to inactivate a context to improve speech recognition accuracy. However, preliminary tests showed very little loss of accuracy when both recognizers were in use at once.

6.4.2 Continuous Recognition With or Without Prompts

In the alternate version of QRSLT/ELSIE that preceded Version 0.6, recognition of an utterance could be initiated either by pressing a button, or by saying the "attention word" (QRSLT/ELSIE) and waiting for the computer to respond with an audible and/or visible prompt that it was ready to receive spoken input. However, tests with users showed that most people would say the attention word only before the first utterance; after that, they expected QRSLT/ELSIE to recognize and translate one sentence after another.

To meet this expectation, the attention word recognition sequence was modified so that in fact all active contexts were consulted when the system was attempting to verify recognition of the attention word. If QRSLT/ELSIE successfully detected one of the known sentences instead of the attention word, the sentence was translated and the output spoken without any prompt. This allowed the user to speak one sentence after another in a normal conversational sequence.

The drawback to this continuous recognition was that the attention word recognition in VTAF took place in a loop that timed out and was restarted over and over again. If a sentence was

spoken just as one loop terminated and another began, QRSLT/ELSIE could fail to recognize the sentence successfully. Using the attention word still yielded the greatest recognition accuracy. The technical issues relating to attention words and the speech recognition loop are discussed in greater detail in the next section.

In keeping with the "hands-free" usage philosophy of Version 0.6, QRSLT/ELSIE no longer displayed a modal dialog box when it failed to recognize a sentence. The error message appears (as it does currently) on the GUI in the edit boxes which would usually contain the recognized input sentence and the translation, but execution of the program continues without any need for user intervention.

6.4.3 Technical Issues Encountered

Side Effects of the "Initial Noise" Parameter -- The ICSS engine had an optional parameter which, when set for a recognition context, allowed successful recognition of sentences even if they began with a "noise" word such as "uh". This seemed like a useful feature, but it was found that it interfered with the recognition of sentences that began with similar sounds. In particular, it negatively affected recognition of the command words "load" and "unload" at the beginning of an utterance. As a result, this feature was not used in Version 0.6 of QRSLT/ELSIE (see Footnote 2).

Unreliability of the "Attention Context" – As noted previously, the ICSS engine allowed the creation of a one-word recognition context called an "Attention Context". When the Attention Context was active, the recognition engine would wait until it recognized this word and then enter a state of readiness to receive further input. But in practice it became apparent that the Attention Context was too sensitive in its recognition. Sometimes users would have to speak the attention word several times before the system would "wake up".

It was found that better results could be achieved by putting the attention word, along with all the other command utterances, into a "Command Context" and calling the ordinary time-based recognition function in a loop until the attention word or some other utterance was recognized. As described in the previous section, this not only improved recognition of the attention word, it also made possible the continuous translation of sentences without the necessity for using the attention word at all.

Real-Time Speech Recognition versus File-Based Recognition – The ICSS system allowed the programmer to choose between recognizing an utterance as it is spoken or recording it in a Microsoft Windows "WAV" format file and recognizing from the file. The file-based recognition is slower, but it has certain advantages. For example, threshold-setting can only be done from a file. Also, once an utterance has been recorded, it can be tested against a large number of recognition contexts sequentially without suffering the loss of accuracy inherent in having many contexts active simultaneously. Utterances could also be tested against the same context multiple times with different recognition parameters.

While Version 0.6 of QRSLT/ELSIE had the ability to do both real-time and file-based recognition, the file-based functions were used only for setting the audio input level threshold. Later versions of the system expanded the use of file-based recognition, for accuracy testing, and other functions, as described in the following sections.

Possible Incompatibility with Windows NT -- QRSLT/ELSIE was written for the Microsoft WIN32 Application Programming Interface (API), to run under both 32-bit Microsoft operating systems, Windows 95 and Windows NT. The alternate version of QRSLT/ELSIE that preceded

Version 0.6 did run equally well under either OS. Preliminary tests of Version 0.6 indicated that the ICSS speech recognition engine was causing compatibility problems under Windows NT.

The source of the problem appeared to be a particular function call within the ICSS API that caused the speech recognition engine to relinquish the sound card temporarily so that another software subsystem could use it, e.g. for sound output. (The issue of sound card control was addressed at length in the preceding discussion concerning the previous version of QRSLT/ELSIE.) Under Windows 95 the function call worked perfectly, but under Windows NT the function call appeared to put the speech recognizer into a suspended state.

6.4.4 Speech Recognition Issues in Spanish

Difficulties with Phonetic Encoding of Spanish – In order to demonstrate two-way translation, it was necessary to bootstrap a Spanish recognizer from the ICSS-VTAF English recognizer. As is the case with most ASR systems, the ICSS system allowed new words to be created by entering phonetic spellings for them in the dictionary. This feature was used to create the Spanish language contexts in QRSLT/ELSIE. However, the phonetic alphabet used in ICSS was not flexible enough to represent accurately the subtle pronunciation differences between English and Spanish. As a result, certain Spanish words were more difficult to recognize than others. The single-word utterance "Hola" (Hello) was the worst offender. If the user pronounced this in a flat English-style (Oh La) the recognition succeeds, but a more genuine Spanish pronunciation will often yield an incorrect recognition of "load all" or "form a". Adding multiple pronunciations for "hola" with different vowel sounds to the dictionary produced some improvement, but the problem persisted. The only long-term solution is to use a speech recognition engine that has been trained for native Spanish speakers, but none were available to us at this point in the QRSLT development.

6.4.5 Problems with Mixing Synthesized and Recorded Output.

Version 0.6 of QRSLT/ELSIE had the ability to "speak" output sentences either with synthesized speech or recorded speech. This dual-mode output capability caused some problems in the volume setting for amplified speakers. The maximum volume level for synthesized output is well below the level of the recorded Spanish sentences in the military/law enforcement context. If the volume is turned up high enough for the synthesized speech to be understood clearly, the recorded sentences will be too loud. Eloquent Technology increased the range of output in later versions of the Text-to-Speech module in order to solve this problem.

6.4.6 Changes in Program Structure

Improved Data Encapsulation. Both earlier versions of QRSLT/ELSIE were designed around the idea that the three major functions of the program, recognition, translation, and output, should be separate entities. The Version 0.3 of QRSLT/ELSIE that used the Dragon speech recognition engine accomplished this separation by dividing the application into three separate programs invoked from a batch file.

The alternate version of QRSLT/ELSIE on which Version 0.6 is based took the object-oriented approach of creating separate C++ class objects to handle recognition, translation and output. However, the nature of the data that the program needed to access led to compromises in the encapsulation of the class objects. For example, the Spanish output sentences were recorded as WAV files that had to be read by the Speech Output object. The input sentences, their

translations and the names of the WAV files containing the translations were stored in a text file, and it seemed logical for the Translation object to read that file. The Speech Recognition object needed to access context files in its own .CTX format. Thus, each of the three major objects had to read files and keep track of data that needed to be synchronized with the data in the other two objects.

The solution to this problem incorporated into Version 0.6 was to create a fourth Context Manager object that does all input of data files and maintains the master list of recognition contexts and input and output sentences for display in the main dialog box. The Context Manager handles all requests for loading and unloading recognition contexts and in turn tells the Speech Recognition object which contexts should be active at any given time. When the contexts are loaded, the Context Manager sends the input sentences and their translations one at a time to the Translation module, which stores them internally without having access to the files from which they originated.

Segregation of Speech Output Functions into a Separate Dynamic Link Library. The Speech Output object still has the unpleasant necessity of keeping track of a growing number of WAV files used for recorded output. To ease this record-keeping burden, the Speech Output object was converted into a separate Dynamic Link Library (DLL) with all of its WAV files compiled into it as resources.

The Speech Output object knows what is recorded in each of its WAV resources and can choose to play a WAV resource or produce synthesized output as needed. No other module in the program knows anything about the WAV resources that are internal to the Speech Output object.

Indexed Lookup for Sentence Translation. The translation module in the previous version of QRSLT/ELSIE used a simple array of character strings to hold the input sentences and their translations. Searching of this array was done in a linear fashion without any attempt at performance optimization. In Version 0.6 this simplistic data structure has been replaced with an indexed (hashed) lookup scheme.

Although later versions of QRSLT/ELSIE perform parsing and translation of input sentences, the sentence lookup table mechanism has remained in place as a preprocessor to handle commonly used utterances quickly and efficiently.

Low-Level Control of Sound Output. The ICSS speech recognition API had a built-in function for playing WAV files. In the previous alternate version of QRSLT/ELSIE this function was used to play the recorded Spanish-language output. When English synthesized speech was added, the ICSS WAV-playing function was still used; output was written to a file by the Text-To-Speech subsystem, then the ICSS function was called. The advantage of this method was that the speech recognizer never had to give up control of the sound board.

In Version 0.6 the Speech Output object controlled sound output at the lowest level allowed in Windows with "waveOut" functions. This allowed the program to control the sound output level and attempt to equalize the volume of synthesized and recorded sound. The disadvantage is that ownership of the sound card became an issue. Although the ICSS subsystem had the ability to relinquish control of the sound card through the ICSSCloseMic() function, this seemed to be the cause of an incompatibility with Windows NT, as described above.

6.5 QRSLT/ELSIE Version 0.9

6.5.1 Design Goals

The consortium members agreed on the following goals for features to be incorporated into Version 0.9 of QRSLT/ELSIE:

- Addition of Mandarin Chinese capability – Version 0.3 of QRSLT/ELSIE could recognize a set of English sentences and output their translations in Spanish. The 0.6 Version added the ability to recognize Spanish sentences and output translations in English. However, feedback from potential users of the system indicated that the addition of other languages would be highly desirable. The consortium members agreed that the Mandarin dialect of Chinese would be chosen as the next language to be incorporated into QRSLT/ELSIE. (The runners-up were Korean, which was added later, and Arabic.)
- Greater Flexibility in Recognition and Translation – Versions 0.3 and 0.6 of QRSLT/ELSIE relied on recognition of entire sentences and subsequent translation through simple table lookup. It was agreed that the 0.9 version should have greater flexibility in both the Speech Recognition and Sentence Translation modules. The table lookup capability would still be present, but the system should be able to parse unfamiliar input and decide whether or not it was close enough in meaning to one of the existing sentences to trigger the same translation. The system should also have the ability to insert variable information, such as numbers or dates, into known utterances. Ultimately, in a later version, the system should be able to parse and translate almost any well-formed utterance within the restrictions imposed by the limited vocabulary of the speech recognition context.
- Low-level Control of Sound Input – The earlier versions of QRSLT/ELSIE used the direct audio input capabilities of the IBM ICSS speech recognition system. Since the IBM ICSS system was eventually to be replaced with Entropic's HMM toolkit (as well as the successor IBM commercial speech recognition product, Via Voice) it was agreed that an alternate method of input would have to be developed. The ability to control sound input at the hardware driver level would give the program the ability to distinguish more intelligently between ambient noise and translatable input, and would also allow more sophisticated processing of recorded utterances.

Initial development work on Version 0.9 focused on the first goal, the addition of Mandarin Chinese to the English and Spanish languages already present. However, the addition of Mandarin required changes to the Sentence Translation module, the Speech Output module, the Context Manager module, and the Graphical User Interface. The remainder of this section discusses these changes and their impact on the later versions of QRSLT/ELSIE.

6.5.2 Changes to the Speech Output Module

For the first implementation of English-to-Mandarin translation we used the same dialog contexts already in use for the English-to-Spanish translations, one dialog context drawn from a law enforcement scenario and one from an emergency medical scenario. New WAV files were recorded for the Mandarin translations and compiled into the Dynamic Link Library containing the Speech Output functions of QRSLT/ELSIE.

As discussed in a previous progress report, the speech output functions were separated from the rest of the program for the sake of modularity and data encapsulation. The addition of the Mandarin sentences verified the success of this modularization. No changes to the Speech Output module were necessary except adding the new WAV files to the list of resources to be compiled and adding the translation strings for each WAV file to a header file.

6.5.3 Changes to the Context Manager

The previous section detailed the addition of a Context Manager module to QRSLT/ELSIE to manage the flow of data between the Speech Recognition module, the Graphical User Interface, and the Sentence Translation module. The addition of Mandarin necessitated changes to the Context Manager similar to those in the lexicon of the Sentence Translation module. The system now has to keep track of the language for each source sentence and its translation.

The Context Manager stores its data in a pair of data structures with the following definitions:

```
typedef struct SentencePairtag  {
      char *SourceSentence;
      char *TargetSentence;
      SentencePairtag *NextSentence;
} SentencePair;

typedef struct Contexttag  {
      long context_handle;
        char *ContextName;
      char *ContextDescription;
      BOOL IsLoaded;
      int SourceLanguage;
      int TargetLanguage;
      SentencePair *FirstSentence;
      SentencePair *LastSentence;
} Context;
```

The first of these structures, "SentencePair", defines a linked list node for storing sentences and their translations. The second structure, "Context", records information about each recognition context including the context's name, a short description to be displayed in a list box when contexts are chosen, and the numeric handle by which the Speech Recognition module refers to the context. The "Context" structure also has a Boolean variable "IsLoaded" to indicate whether or not it is currently active, two variables to record the source and target languages of sentences managed by the context, and pointers to the head and tail of a list of the previously defined "SentencePair" nodes.

Although the Context Manager maintains a list of sentences and their translations, it does not perform lexical lookup when translation is done. The Context Manager stores the sentences for the purpose of displaying them in a list box in the Graphical User Interface, but it hands off the job of translation to the Sentence Translation module by calling that object's "AddLexEntry" member function as it reads each sentence pair from its data files. Here is the prototype of that function:

```
BOOL AddLexEntry(char *src, char *tgt, int src_lang, int tgt_lang);
```

Once the sentences are stored by the Sentence Translation module in its lexicon, the Context Manager no longer needs to refer to them except when the GUI list box containing the source sentences is refreshed.

At this point in the development, the Context Manager did not prevent the user from loading an English-to-Spanish and English-to-Mandarin version of the same recognition context. In this case the output language for a given English input sentence would be that of the first context in which the Speech Recognition object found the sentence. In a later version, a "sanity check" was added to prevent the user from getting unexpected results if duplicate contexts are loaded accidentally.

6.5.4 Implementing Low-Level Control of Sound Input

6.5.4.1 Disadvantages of the VTAF Sound Input System

The early versions of QRSLT/ELSIE used IBM's ICSS/VTAF continuous speech recognition system for the ASR component, on the assumption that this was a temporary replacement for Entropic's Hidden Markov Model Tool Kit, which was in the process of being ported to the PC platform throughout most of the project. Audio input was done by letting the ICSS/VTAF Recognizer take direct control of the PC's microphone up to this point in the development.

This approach to speech recognition had several disadvantages, not the least of which was that we did not expect to be using ICSS/VTAF throughout the development, but assumed that we would have to switch to a new input method when the PC version of HTK became available. Quite apart from that, however, there were several other factors which made the direct audio control by ICSS/VTAF undesirable:

- **"Attention word" recognition performed poorly.** ICSS/VTAF had an "attention word" mode in which the system remains quiescent until the detection of a pre-selected word "wakes it up." In practice, this gave poor results. In addition, feedback from users of the system indicated that most users wanted continuous recognition and did not want to use an attention word before each utterance.
- **Continuous recognition had to be simulated through time-outs and restarts.** Apart from the attention word mode, ICSS/VTAF had no input mode in which it listened continuously for input. Each speech input cycle has a maximum duration of 8 seconds. The best we had been able to do is to simulate continuous recognition by restarting the input cycle each time a cycle times out. This worked significantly better than the "attention word" mode, but it was prone to error because utterances that began during the time-out/restart sequence were often lost or misinterpreted.
- **The algorithm for detecting the beginning of an utterance was inadequate.** The ICSS/VTAF system assumed that an utterance had begun if the input level rose above a certain threshold. Ambient noise and non-verbal sounds by users would often exceed the threshold, causing false utterances to occur. At best, these utterances would fail silently. At worst, a non-verbal sound would be "recognized" (and subsequently translated) as if it were a genuine utterance. The system did not attempt to distinguish between speech and noise before invoking the speech recognition mechanism.
- **Setting of the input threshold often failed.** There was a function in the ICSS/VTAF API for automatically setting the input threshold described above, but this function

often failed. The failure appeared to be hardware dependent. It was more prone to failure on some machines, and there were some machines on which the function always failed.

- **Utterances could not be saved for later reprocessing.** It was desirable to save utterances as Windows-format WAV files as they were spoken. These saved utterances could then be processed further, for example by putting them through a noise reduction stage, and then speech recognition could be retried. Saved utterances would also be useful for off-line batch processing to test speech recognition parameters and post-processing methods. The ICSS/VTAF system had a mechanism for saving files and for recognizing from WAV files, but the file saving mechanism was as prone to failure as the threshold setting function.

6.5.4.2 Speech Recognition versus Spoken Language Processing

Experience with early versions of QRSLT/ELSIE, especially the experience of demonstrating the system to users unfamiliar with speech recognition technology, had shown that a "Quick Response Spoken Language Translator" must go beyond the traditional boundaries of speech recognition in processing audio input. Most speech recognition systems (and the Application Programming Interfaces used to program them) assume that the beginnings and endings of utterances will be clearly marked, and they concentrate solely on decoding the contents of such utterances. In fact, a higher level of processing which we will call "Spoken Language Processing" must precede the speech recognition stage. A "Spoken Language Processor" must be capable of monitoring sound in real-time and making intelligent decisions as to whether or not a given set of input samples contains relevant human speech that should be translated or noise that should be discarded. Note that the "noise" category could also include speech, such as voices in the background or an utterance that is aborted when the system user and another person nearby begin speaking at the same time.

The differences between Spoken Language Processing and Speech Recognition can be compared to the differences between continuous and discrete speech recognition: many of the methods and mathematical tools are the same, but they must be applied differently.

Even after an utterance has been detected and its words have been recognized, another processing step may be necessary to "complete" the utterance by adding words which are missing either because they were left unspoken by the system's user or because the recognizer failed to decode them. For example, we have found through experience that many utterances fail to decode properly because either the first or last word of the utterance was lost or misinterpreted by the recognizer. This "utterance completion" stage spans the boundary between the speech recognizer and the translation module.

The realization that continuous speech recognition in the ASR component in itself is insufficient for the purposes of a Quick Response Spoken Language Translator brought about a revision in the conceptual model of the program. The original model depicted three semi-autonomous processing stages:

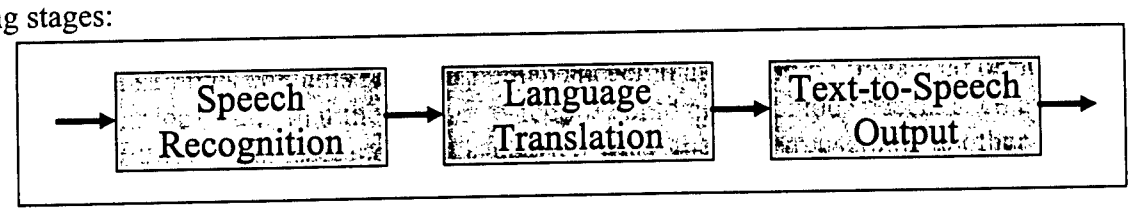


Figure 6.5.1: Original conceptual model of Spoken Translation system.

A more complex conceptual model, which includes the concept of Spoken Language Processing, adds more detail to the first processing stage and shows the interconnection between this stage and the Translation module:

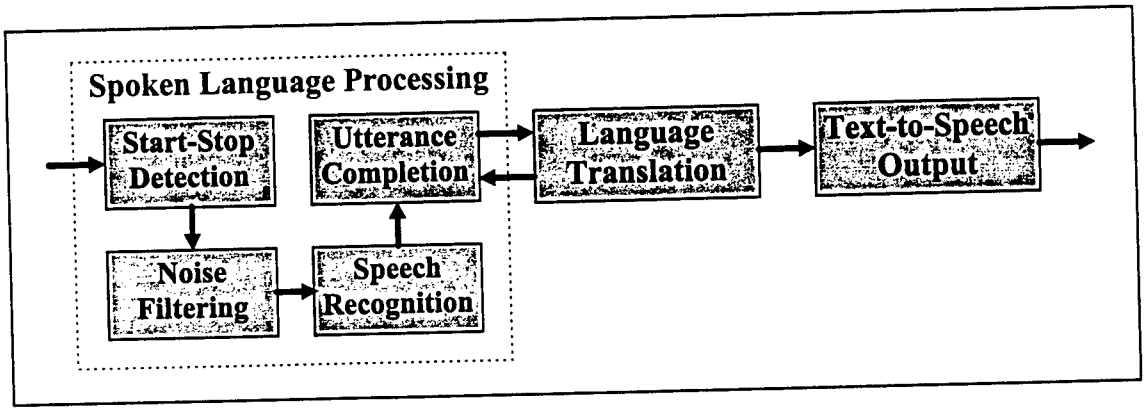


Figure 6.5.2: New conceptual model of Spoken Translation system.

The following sections detail how the audio input system was redesigned so that we could effectively implement the spoken language processing functions shown above.

6.5.4.3 Goals of the New Audio Input System

The overall goal in designing the new audio input system was to achieve greater control over sound input. Rather than passing control to an autonomous "black box" subsystem and waiting for it to return a string of recognized text, we wanted to be able to make our own processing decisions at each stage of audio input. Here are some specifics of this greater level of control:

- **Continuous input without periodic restarts.** When the system is running in its automatic mode, the microphone should be ready to pick up speech at all times. This requires continuous processing of input buffers received from the sound card. It also requires multi-threading with a high priority given to the sound input thread so that monitoring of the microphone can take place even while previous utterances are being recognized and translated.
- **Saving of files for reprocessing or off-line testing.** Although the system must demonstrate "quick response" to speech in real-time, the utterances it receives should also be saved in files. This would allow later reprocessing of utterances either to improve recognition or refine discourse-level understanding of the conversation with the user. The saved files are also useful for off-line testing of speech recognition components.
- **Ability to edit sound input in real-time.** It should be possible to disassemble and reassemble portions of apparently separate utterances when the need arises. For example, an utterance may be interrupted by a long pause or some extraneous noise. Typically this causes a failure in recognition and translation because the system hears

the input as two separate utterances. If the noise or pause can be deleted and the two utterances joined into one, then recognition and translation can succeed.

- **Intelligent algorithm for analysis of utterance before recognition.** Under the current system, much processing time is wasted because a recognition cycle is attempted every time the sound input level goes above a predefined threshold. As mentioned earlier, in the worst case a bit of random noise may be mistaken for a valid utterance. The "noise filtering" step shown in the diagram above should attempt to differentiate between speech and non-speech without using as much CPU time as a full speech recognition cycle.

6.5.4.4 Changes in Program Structure

The goals listed above required a restructuring of the existing QRSLT/ELSIE program. This section describes the technical details of this restructuring and the impact that the changes have had on program flow.

The principal effect of the restructuring was that the program was more "Event-Driven" than before. When we were using ICSS/VTAF for sound input, we could rely on synchronous API calls to keep the program flow linear. In other words, when we called the function that initiated a "live mike" recognition session, we could depend on the fact that the recognition session would be complete when the next instruction was executed. This was no longer be the case. When the program's controlling thread wants to turn on the microphone and start listening for input, it must do so by sending a message to the execution thread controlling the sound card; it must then be prepared to wait indefinitely until the sound card thread sends back a message saying that an utterance has been received. In the meantime it may be called on to perform other tasks, such as updating the user interface. Figure 6.5.3 below illustrates the flow of information during a typical cycle of input, recognition, translation and output.

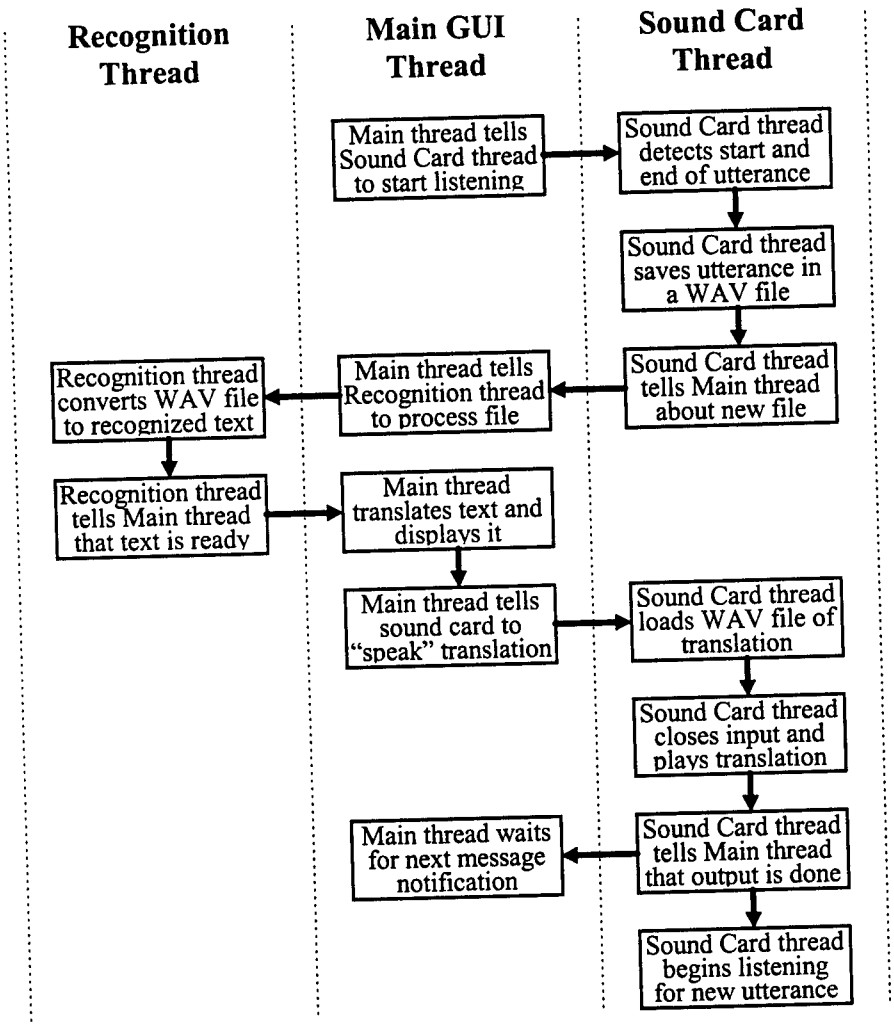


Figure 6.5.3: Information flow during input, recognition, translation and output cycle. (Each arrow crossing a dotted line represents a Windows Message)

QRSLT/ELSIE was already using multi-threading to prevent the system from becoming unresponsive during the speech recognition and text-to-speech synthesis cycles, but it was spawning short-lived "worker threads" as needed. After the restructuring, all sound card functions were performed by a permanent "window thread." The purpose of using a "window thread" for the sound card is to allow it to send and receive messages, which a "worker thread" cannot do; an actual GUI window is associated with the thread but remains invisible.

A side effect of the program reorganization was that the sound output functions, which previously had been separated into their own Dynamic Link Library, had to be reintroduced into the main executable program so that the Sound Card thread could control both input and output. However, for convenience and modularity the WAV output files themselves remain segregated in their own DLL.

6.5.4.5 Additional Design Considerations

The program reorganization described above raised many design questions that can potentially affect system performance:

- **How large should the audio input buffers be?** In performing low-level audio input we must prepare data buffers for the operating system and add them to a buffer queue. After a buffer is filled with 16-bit sound samples, the operating system sends a window message to let the program know that it should process the buffer. The frequency with which this happens is controlled by the size of the buffers. The issue here is granularity. If the buffers are too large, then the program's response time will seem slow because it will be unable to respond to rapid changes in sound level. (The program will not be aware of a change until an entire buffer has been received and processed.) If the buffers are too small, unnecessary overhead will be added by the necessity of frequent context switching to process them. In the initial version we have chosen a buffer size of 2205 samples, so that the sampling time is exactly 200 milliseconds. (The sampling rate is 11025 Hz.)
- **How many files should be saved?** The Sound Card thread saves audio input in WAV files and notifies the Main GUI thread when a file is ready to be recognized and translated. It would be desirable to maintain a "ring buffer" of files so that we can save previous utterances for later processing. Here the issue is simply one of disk space. For now we are using a ring of 100 files. The first file saved is called QRSLT/ELSIE_TEMP_00.WAV, and the 2-digit number at the end of the name is incremented for each new file. After 99 it wraps around. On the average the WAV files that are saved tend to be between 10K and 100K, so this means that from 1 to 10 Megabytes of disk space will be required to save these files.
- **When should noise discrimination be done? Before or after saving the file?** There are two possible approaches to the problem of determining whether sound input is a legitimate utterance to be translated or random noise that should be ignored. One approach is to try to make this decision in real-time on a per-buffer basis as each 200 millisecond audio buffer is received. The other approach is to allow the sound input to be saved in a file and then apply a noise discrimination algorithm to the entire file. After experimenting with both approaches we decided to pursue the latter, although some statistical analysis of each buffer is performed in order to adjust the input threshold.
- **Which statistics give the most information with the least processing cost?** This question is intimately connected to the previous one. The selection of an algorithm for noise discrimination is affected by the amount of processing time the algorithm requires. Sophisticated algorithms for processing sound samples require the use of the Fast Fourier Transform, which is computationally intensive. Other algorithms can make use of simple mathematical formulas based on computing means and standard deviations of sample levels within sound buffers. Experimentation will be required to determine how much processing time is acceptable and which algorithm gives the best results within that timing window.
- **What is the optimum threshold? How can this be determined automatically in real-time?** The first level of sound input processing is to decide how large the input

samples must be before we pay any attention to them at all. This requires setting a threshold level. As described previously, the setting of the threshold level was one of the greatest problem areas within the ICSS/VTAF API. Using the statistics we are gathering for each buffer, it should be possible to define an algorithm for adjusting this threshold level continuously. This would allow the system to compensate for changes in ambient noise without requiring user intervention.

6.5.5 Features for Incorporation into the12 Month Benchmark

At the end of the quarter, the system modifications described above were incomplete, so the official 9 month benchmark version of QRSLT/ELSIE still used the IBM ICSS/VTAF API for sound input as well as recognition.

The system features which were under development for the next quarterly benchmark were the following:

- **Self-adjusting Threshold.** QRSLT/ELSIE should be able to adjust its input sensitivity based on a continuous monitoring of peak and average sound levels. This feature is especially important when the system is used by several people with different voice qualities and in environments whose background noise may change frequently.
- **A "Sound Status" Control to guide the user in adjusting the microphone.** Experience with test users has shown that adjusting the microphone input to the proper level is crucial in achieving optimum results from the system. Previously we had depended on an inherently unreliable threshold-setting mechanism to give the user feedback about input levels. Along with the automatic threshold-setting described above, we were working on the implementation of a graphical control to show the user the current status of audio input.
- **Pre-processing and post-processing of recorded input.** QRSLT/ELSIE was being redesigned to save audio input in files while the system is in use. In the long term we would like to be able to achieve a discourse-level understanding of an entire conversation by the post-processing of these files. In the short term, we wanted to be able to discriminate between noise and voice by pre-processing the files before the speech recognition stage.

6.6 QRSLT/ELSIE in Transition from V0.9 to 1.2

The 9 month benchmark (0.9) Version of QRSLT/ELSIE that was distributed to the consortium members at the third quarterly meeting was a transitional version. It had many new features, most notably the introduction of English-to-Mandarin contexts covering the same corpus as the previously defined English-to-Spanish and Spanish-to-English recognition and translation contexts (law enforcement and medical). However, there were several new features in development which were not ready in time for this benchmark version. In particular, the implementation of low-level audio input, discussed at length in the previous section, was still being debugged when the 0.9 benchmark version was prepared. For the sake of stability, the 0.9 Version still used the IBM ICSS/VTAF API for direct audio input.

For the 12 month benchmark, the implementation of low-level audio input appeared to be stable, and it replaced the ICSS "live mike" recognition function in the official version of the QRSLT/ELSIE system. The remainder of this section discusses the details of the low-level

audio input feature as it was implemented for the 12 month benchmark (V.1.2 of the QRSLT/ELSIE system.

6.6.1 Processing of Sound Buffers

A new C++ window class called CWaveWnd, which runs in its own thread, now handled all audio input and output. The CWaveWnd class manages a number of input buffers, each of which can contain 200 milliseconds of sound (2205 16-bit samples). We have set the number of sound input buffers to 70, based on a maximum of 10 seconds of continuous recording (50 buffers) plus 20 extra buffers to allow the system to continue to receive input while a maximum length utterance is being saved to a file.

When sound input begins, these 70 buffers are prepared and submitted to the operating system through calls to the "waveIn" family of low-level audio functions that are part of the standard WIN32 API. From then on, as long as recording continues, the operating system sends a callback function every 200 milliseconds telling the program that a particular input buffer is full and ready to be processed.

Processing of the input buffers is performed by the "OnProcessWavBuf()" function. This function computes some simple statistical values from the 16-bit samples in the buffer and decides how the buffer will be handled. If no utterance is in progress but the statistics indicate that the buffer contains samples that exceed the input threshold, the processing of a new utterance begins. Conversely, if utterance processing is in progress but analysis of several consecutive buffers shows no further sound input taking place, the buffers containing the utterance are saved to a file and the main processing thread is notified that a new input file is ready for speech recognition and translation.

In deciding whether or not a sample buffer contains part of an utterance, two different metrics are applied. The simplest is to determine whether the absolute magnitude of any sample has exceeded the input threshold level. (The computation of this threshold level is discussed below.) As a secondary check, the average difference between successive samples in the buffer is calculated. If one or more samples has exceeded the threshold but the average difference between consecutive samples is very small, the samples that exceeded the threshold were probably a "noise spike" that can be ignored.

This utterance determination method has the potential defect that it can mistakenly ignore a buffer in which an utterance begins near the end of the buffer. To prevent this from occurring, the point at which buffers are marked for saving to a file is set to begin two buffers prior to the detected start of the utterance. This has also been very effective in preventing the clipping of utterances which begin with soft aspirated sounds that may not exceed the input threshold.

After each buffer is processed, it is "unprepared" (Microsoft's terminology) and then prepared again and resubmitted to the operating system for reuse. The 70 input buffers are used continuously in a ring, so that buffer number 0 is used again after buffer number 69.

6.6.2 Automatic Setting of Input Threshold

The OnProcessWavBuf() function records statistical values that it computes for each input buffer. At a regular interval, which currently is set to every 5 buffers (1 second), this function calls the DoBufferStats() function to examine the statistics and adjust some of the parameters

governing the audio input process. It is here that the automatic adjustment of the input threshold takes place.

We have experimented with a number of different statistical measures for setting the input threshold. The one we are currently using, which has proven to be the most effective, is to record the maximum range (difference between the largest and smallest input samples) for each buffer. If an utterance is in progress, this range will be large and will represent the magnitude of speech input. If no utterance is in progress, this range will be small and will represent the magnitude of the background noise level due to a combination of ambient sound and noise introduced by the microphone.

For each one second statistical period the maximum buffer range and minimum buffer range are computed. Then a new threshold value is computed based on the following algorithm: pick whichever is larger, the maximum range divided by a constant or the minimum range multiplied by a constant. This assures that the threshold will stay well above the level of background noise but below the level of utterances.

This new computed input threshold is not substituted directly for the old threshold value. Rather, the old value is adjusted by a certain percentage of the difference between itself and the new value. This prevents abrupt changes in threshold. The percentage of adjustment decreases gradually as the input session continues, so over time the threshold will find its proper level and remain relatively constant.

6.6.3 New Functionality of the "Set Threshold" Command

In previous versions of QRSLT/ELSIE there was a "Set Threshold" command which a user could select to initiate a call to the ICSS/VTAF threshold setting function. This function required the user to speak a test sentence, after which a new threshold setting was computed and put into place. In practice, however, this function often failed.

In the new version of QRSLT/ELSIE the "Set Threshold" command has been retained, even though adjustment of the threshold is now automatic and ongoing. The new command tells the user the current threshold level and then initiates a recalibration sequence. In this recalibration the percentage of threshold change allowed for each one-second interval is reset to the maximum allowed when the session was initiated; in other words, the threshold is allowed to seek its new level more quickly. Then, over time, the percentage of change allowed decreases gradually as before to minimize sudden threshold changes.

6.6.4 Graphical Indication of Input Level

Once we had a mechanism in place for computing current sound levels and adjusting the threshold, we decided that it would be advantageous to communicate this information to the user. Our experience in demonstrating QRSLT/ELSIE on various computers with different microphones has shown that the adjustment of the input level is crucial to getting good performance from the system. Unless the user can see the input level, he or she may not be aware that an adjustment is necessary.

In the new version of QRSLT/ELSIE, the following graphical control is displayed when sound input is started:

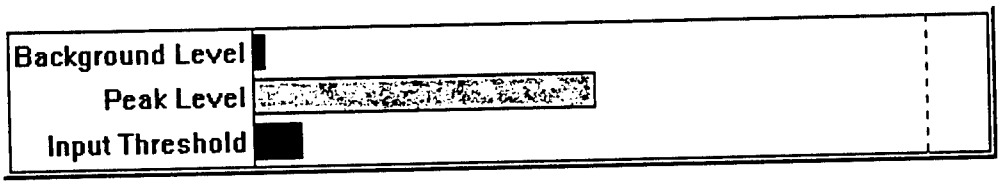


Figure 6.6.1: QRSLT/ELSIE's Sound Input Level Indicator

The bars on this graph are updated every second as the DoBufferStats() function computes its statistics. The "Peak Level" bar is normally displayed in green, but its color changes to red if it extends beyond the dotted line on the right of the graph, indicating saturation of the microphone.

The optimum performance of the system occurs when the peaks during speech reach just beyond the halfway point on the graph. If, on the other hand, the peaks never exceed the input threshold level, then the system will appear to be "dead"; with this graphical indicator, the user will understand that the system's unresponsiveness is due to the lack of sufficient input signal, not to a malfunction in the program.

6.6.5 Directions for Further Development

Testing QRSLT/ELSIE with many different types of PC and many microphones has shown us how important the quality of sound input is for overall system performance. Unfortunately, we may not be able to control this variability in input devices unless the system is marketed as a combination of hardware and software with a sound card and microphone that have been shown to perform well. Many manufacturers of speech recognition systems take this approach and refuse to support hardware that has not been tested and approved.

Regardless of what kind of sound hardware is present, the system can perform poorly if the operating system software that controls the microphone input level is set incorrectly. This problem shows up most frequently when switching between different microphones on the same PC. If an unpowered microphone is used, the operating system's "Volume Control" mixer setting for the microphone usually must be set near the maximum in order to get sufficient input. If an amplified microphone is used at this setting, it will saturate the input and the system will not function properly. With amplified microphones we have found that a "Volume Control" mixer setting of one quarter to one third of the entire range is usually the best.

This problem of software setting of input level is complicated by the fact that the use of the operating system's "Volume Control" applet is not at all intuitive. When the applet runs it always displays playback settings only; changing the settings for recording levels can only be done after making the correct selection from an "Options" pull-down menu. In future versions of QRSLT/ELSIE it may be necessary for the program to present its own mixer interface to the user or even to implement automatic setting of the mixer for recording input level.

Another interesting facet of sound hardware on various systems is that more and more PC's now come equipped with sound cards that are "full duplex". These sound cards present themselves to the operating system as two separate devices. Thus, recording and playback can take place simultaneously. The use of a full duplex sound card would represent a significant simplification in the task of controlling audio input and output. On a half-duplex system the sound input device must be closed before sound output can be played, and the sound output device must be closed before sound input can resume. Since QRSLT/ELSIE performs audio input and output with each

utterance, a lot of time and effort is spent in opening and closing the sound card and making sure that input and output are properly synchronized.

A possible improvement to the program might be to have QRSLT/ELSIE determine whether or not its sound card is full duplex and act accordingly. This would increase the complexity of the program slightly but could offer a noticeable performance improvement on systems that have the proper kind of sound hardware.

6.6.6 Distribution of an 11 month Benchmark

The previous two sections describe a major revamping of the audio input module of the QRSLT/ELSIE program. Since this module appeared to be stable and was performing well at this point, a "Version 1.10" (11-month) benchmark version of the program was prepared and distributed over the Internet to the other consortium members, replacing the 9-month version distributed at the end of the previous quarter. Also, for the first time the complete source code for the QRSLT/ELSIE user interface developed by LSI was made available to the consortium in order for Entropic to develop an interface for their HTK speech recognition system to replace the IBM ICSS/VTAF recognizer which had been used as an interim solution.

6.6.7 Summary of Speech Recognition Milestones

The focus of LSI's development effort on the speech recognition component of QRSLT/ELSIE shifted to a new area: introducing the ability to recognize the Mandarin dialect of Chinese. The latter topic will be discussed in detail later in this section.

The goal of the Quick Response Spoken Language Translation project was to develop a system that can recognize and translate utterances in two directions in several languages. In the first working versions of the program (V.0.3 and alternate V.0.3: see Sections 6.1 and 6.2), English was the only language recognized. Later the ability to recognize Spanish was added (Section 6.3). The addition of a Spanish recognition capability was accomplished by adding Spanish words phonetically to the existing English-language dictionary of the IBM VTAF recognizer, not by starting from the ground up and constructing a Hidden Markov Model recognition system from a training corpus of recorded Spanish-language speech.

In spite of the fact that the Spanish speech recognition was bootstrapped from the English recognizer, the performance was adequate. The constraints imposed by recognizing complete sentences and limiting the number of utterances were sufficient to allow the system to recognize almost all the sentences in the Spanish version of the law enforcement and medical contexts previously defined for English. The most problematic sentences were one-word utterances (e.g. "Hola!").

We employed the same method to introduce Mandarin speech recognition. In this version of the system, QRSLT/ELSIE was able to recognize (via the VTAF recognizer) and translate the sentences in the law enforcement and medical contexts in both directions between English and Mandarin as well as English and Spanish. The following section discusses some of the details of the implementation of Mandarin recognition.

6.6.8 The Search for Phonetic Equivalents

The IBM ICSS/VTAF speech recognizer, which we were using as an interim solution while Entropic ported its Hidden Markov Model Tool-Kit to the PC platform, allowed new words to be

created by entering phonetic spellings for them in the dictionary for the speech recognizer. However, the phonetic spellings are restricted by the ICSS/VTAF system's repertoire of English sounds.

To encode words in Mandarin, we established a correspondence between the phonetic system of Mandarin and the phonetic system of English, using the ICSS/VTAF English phonetic repertoire. The following table is a list of the phonetic equivalents. The first column of this list contains the vowels and consonants in Mandarin, as well as the frequent combinations of vowels and consonants, in Pinyin format. (Pinyin is the standard alphabetical representation of Mandarin used in Mainland China). The second column of the list is the phonetic representation of these sounds using the ICSS/VTAF convention:

Pinyin	ICSS/VTAF
A	AA
I	IY
O	AO
U	UW
U:	UW
E	AX
AO	AW
OU	OW
UO	W AO
AI	AY
EI	EY
IE	Y EH
UI	W EY
IU	Y OW
IAN	Y AH N
IAO	Y AW
UAN	W AA N
UE	Y W EH
ER	ER
AN	AA N
EN	AX N
IN	IY N
ANG	AA NG
ENG	AX NG

Pinyin	ICSS/VTAF
ING	IY NG
ONG	OW NG
B	B
P	P
M	M
F	F
D	D
T	T
N	N
L	L
G	G
K	K
H	HH
R	R
J	JH
Q	CH
X	SH
Z	DD Z
C	TS
S	S
ZH	JH
CH	CH
SH	SH
W	W
Y	Y

An obvious limitation of using this bootstrapping method is that we had to use the ICSS/VTAF phonetic system designed for English to represent the sounds in Mandarin. Although most Mandarin sounds cannot be represented accurately by English sounds, many can be approximated by English sounds and could be recognized by the ICSS/VTAF speech recognizer.

6.6.9 Problems Encountered in Adapting ICSS/VTAF for Mandarin

The main problem that we encountered in sound representation was that the ICSS/VTAF phonetic repertoire was too limited to represent the sounds in Mandarin, even with the approximation method. There are some sounds in Mandarin that are far too different from any English sound to be represented by the ICSS/VTAF phonetic system.

Some of these sounds have similar sounds in other languages such as French. "U:" is such a sound. The ICSS/VTAF phonetic alphabet did include some sounds from other Romance languages to help in recognition of foreign words that are commonly used in English, so we were able to select the phonetic representation for this sound in French and use it to represent "u:" in Mandarin. There are some other Mandarin sounds that we had to represent with a sound in English which is as similar as possible, such as "zh" (represented by "jh" for English).

A more serious problem in sound representation is that in two cases we were forced to use the same phonetic representation for two different sounds in Mandarin. Thus, both "j" and "zh" were represented by "jh" in our dictionary, since "jh" is the closest English sound we could find for both of them, and there was no way for us to capture the difference between them using the limited ICSS/VTAF phonetic repertoire. The other such case is with "sh" and "x" in Mandarin, both represented here by "sh" for English.

One of the main challenges for a Mandarin speech recognition system is to recognize tones. Mandarin is a tonal language. There are four basic tones in Mandarin, plus a fifth neutral (or light) tone. Tones are phonemic in Mandarin, meaning that they are used to distinguish different words with the same consonant and vowel combination. For example, the word "dui4" in our corpus has the meaning "line", and "dui1" (not in our corpus), which differs from "dui4" only in tone, is a different word meaning "pile". In the 1.2 version of QRSLT/ELSIE, we did not deal with tone recognition at all, since the ICSS/VTAF speech recognition system was not capable of recognizing tones. Although in our dictionary we marked the words with tones, the ICSS/VTAF system recognized Mandarin words (and sentences) without using this information.

Another major challenge for Mandarin speech recognizers is to handle the problem of homonyms. There are many homonyms in Mandarin, i.e. words that are pronounced exactly the same with the same vowel and consonant combination as well as the same tone. The problem of homonyms was not yet serious at the current stage of project development because there were few homonyms in the small Mandarin dialog corpus we were using at this time. The only homonyms that we had in that corpus were the words "shi4" ("matter") and "shi4" ("be"). This particular case was not a problem at this point, since the speech grammar at that stage of development was written in such a way that we were recognizing whole sentences instead of individual words, and the homonyms were distinguishable in terms of their context. In the future development of QRSLT/ELSIE, when we have more sophisticated speech grammars and are recognizing individual words

and phrases, the homonym problem will be more prominent. In that case, a possible solution still is to make use of the context to disambiguate the homonyms.

6.6.10 Performance of the Mandarin Recognizer

Considering the sound representation problems, our system performed fairly well in Mandarin speech representation, with about 90 percent recognition accuracy. The recognition of some sentences was very good, in that the system never failed to recognize them. Other sentences needed a few tries for the system to get them right. (However, this is an example of the user training herself to the system and not of the system "learning" to recognize the sentences correctly, since the ICSS/VTAF recognizer does not change its recognition parameters over time.)

As with Spanish, there are a few short sentences that the system simply cannot recognize unless the speaker deliberately uses "English-like" pronunciation. For example, the system could not recognize "*Xie4xie*" ("Thank you") unless the consonant in "*xie4*" is pronounced "sh" as in the English word "*shave*." This was not surprising, however, since we were using "sh" to represent the "x" sound in Mandarin. The "sh" sound, although not correct, is the closest sound we could find from the English consonant repertoire to represent the Mandarin consonant rendered as "x" in Pinyin.

6.6.11 Future Directions

The long-term solution to the problems mentioned above is to use a speech recognition engine that is designed specifically to recognize Mandarin speech. Because the new version of Entropic's Hidden Markov Model Tool-Kit was to have the ability to recognize fundamental frequency (F0), we had hoped that it would be possible for it to distinguish the tones of Mandarin. We had expected a substantial improvement in Mandarin recognition capabilities when the HMM Tool-Kit was incorporated into QRSLT/ELSIE. Unfortunately, the Chinese recognizer was not made available to LSI during the QRSLT project.

6.7 Preparing for the Transition from IBM VTAF to HTK (QRSLT/ELSIE V.15)

Since the beginning of the QRSLT project, as noted previously, the QRSLT/ELSIE program had used an unreleased beta version of IBM's ICSS/VTAF to provide speech recognition capabilities. The consortium agreed on this as an interim solution until Entropic Research Laboratory finished porting its HTK Application Programming Interface (HAPI) from UNIX to the Win32 platform.

In July, 1997, Entropic Research Laboratory sent a pre-release version of HAPI for Win32 to LSI, and the LSI staff devoted a significant amount of effort to learning to use the API, testing the performance of this new speech recognizer, and planning the changes necessary to incorporate HAPI into QRSLT/ELSIE. The remainder of this section covers some of the technical issues involved in this transition.

6.7.1 HAPI and SHAPI

In contrast to the IBM ICSS/VTAF speech recognition API, which is a "black box" system with very few options or modifiable parameters, Entropic's HAPI has dozens of

parameters and exposes many of the inner workings of the recognizer. This means that the programmer can exercise greater control over the recognition process and customize the recognizer for the task at hand; the trade-off is that the learning curve is steeper.

To assist LSI in the use of the HAPI recognizer, Entropic created a simplified version of HAPI (dubbed SHAPI, for Simplified HTK Application Programming Interface). The aim was to give LSI a set of function calls that were as close as possible to the high-level functionality offered by the IBM ICSS/VTAF system.

6.7.2 Rewriting the SpeechRec and ContextMgr Classes

The first task for LSI after receiving the pre-release version of HAPI was to incorporate the new recognizer into the recognition-related modules in QRSLT/ELSIE.

By design, the speech recognition functions of QRSLT/ELSIE had been encapsulated in a C++ class called "SpeechRec" (Section 6.4). LSI had also rewritten the audio input module of QRSLT/ELSIE (Section 6.6) so that microphone input is handled at the lowest Win32 level and all utterances are saved in files, thus reducing dependence on a particular recognizer's methods for controlling the microphone. The purpose of these design decisions was to make the transition from the IBM speech recognizer to the new Entropic recognizer easier.

When the initial version of the new SpeechRec class was ready, it was incorporated into a command-line program that processes batches of WAV audio files and records timing and accuracy statistics.

6.7.3 Testing Recognition Accuracy

For test purposes we prepared lattice files and a small dictionary file containing the 48 sentences from the original Law Enforcement and Medical contexts used by the first version of QRSLT/ELSIE. Then we recorded 16KHz WAV files for all forty-eight sentences in both English and Spanish (using native speakers). The WAV files were given names containing the verbal content of the recording (i.e. the utterance "Move back" was put in a file called "Move back.WAV") so that the test program could compare the results of speech recognition with the name of the file and determine whether or not the recognition was correct.

Initial tests of the English recordings produced excellent results. Using almost all of the default recognition parameters supplied by Entropic, we saw only a single recognition error in 48 utterances (98% accuracy). The only parameter we changed from the default was to force static Cepstral Mean Normalization, an option that can only be used when recognition is being done from a file instead of a live microphone. Further experimentation showed that a slight increase in the WORDBEAM parameter (to 125 from the default of 75) achieved a perfect score of 48 out of 48.

Next we tested the Spanish recordings. To do this, we prepared a parallel version of the test program we had used for English and modified the configuration parameters of the recognizer to use the language model for Spanish that Entropic had provided. We also prepared Spanish versions of the lattice files and dictionary for the 48 sentences.

Initial results with Spanish were poor: 32 out of 48 correct (66.7%) using the same default parameters we had used for English.

6.7.4 *Experimentation with Recognition Parameters*

After getting such different results with the English and Spanish language models for the Entropic recognizer, we began to experiment with recognition parameters to see if the performance on the Spanish test utterances could be improved. We found that recognition accuracy could be greatly improved by increasing the WORDBEAM and GENBEAM parameters together. These parameters affect the "width" of the search by increasing the likelihood range of possible paths through the utterance. The following table shows the steady improvement in performance as the parameters were increased. No further improvement was seen when the GENBEAM and WORDBEAM parameters were increased beyond 550.

GENBEAM	*WORDBEAM*	*# CORRECT*	*TOTAL*	*PERCENT*
150	75	32	48	66.7%
250	250	36	48	75.0%
350	350	41	48	85.4%
450	450	42	48	87.5%
550	550	45	48	93.8%

Table 1 – Spanish Recognition Accuracy (Monophone Model)

The downside to increasing the GENBEAM and WORDBEAM parameter settings is that this causes an increase in computation time. At the point where an acceptable level of accuracy was reached, the recognition of each utterance was taking several seconds.

6.7.5 *Monophone versus Triphone Language Models*

When we presented our results to Entropic, they suggested that we try using a "Triphone" language model for Spanish instead of the "Monophone" model we had used originally. The following table contains our results with the Triphone model:

GENBEAM	*WORDBEAM*	*# CORRECT*	*TOTAL*	*PERCENT*
150	75	30	48	62.5%
250	250	33	48	68.8%
350	350	33	48	68.8%
450	450	34	48	70.8%
550	550	44	48	91.7%
650	650	44	48	91.7%
750	750	46	48	95.8%
850	850	47	48	97.9%
950	950	48	48	100.0%

Table 2 – Spanish Recognition Accuracy (Triphone Model)

Using the Triphone model produced no better results unless we also set the FORCECXTEXP parameter to TRUE. With this model and parameter change, we were able to get perfect results if we made the GENBEAM and WORDBEAM large enough.

These results were encouraging, but the even larger values of GENBEAM and WORDBEAM required made the recognition process even slower.

Still more experimentation proved that we could improve the accuracy and speed of the Triphone model if we also set the FORCELEFTBI parameter to TRUE. This parameter allows the recognizer to use its Triphone speech model within words but forces use of a "Left Bigram" at the ends of words. Here are the results:

GENBEAM	*WORDBEAM*	*# CORRECT*	*TOTAL*	*PERCENT*
150	75	34	48	70.8%
250	250	45	48	93.8%
350	350	45	48	93.8%
450	450	47	48	97.9%
550	550	48	48	100.0%

Table 3 – Spanish Recognition Accuracy (Triphone Model with FORCELEFTBI)

This improvement was encouraging because it indicated that we could get extremely good accuracy with a moderately high GENBEAM/WORDBEAM combination (550) and acceptable accuracy with a relatively small value (250).

6.7.6 Integrating HAPI into QRSLT/ELSIE

After the initial testing phase was complete we prepared a test version of QRSLT/ELSIE using the new "SpeechRec" class that calls Entropic's HTK/HAPI recognizer instead of IBM's ICSS/VTAF. Two separate versions were prepared, one for the English recognizer and one for the Spanish recognizer, because we had not yet solved the problem of how to do recognition of both languages simultaneously using HAPI.

These test versions of QRSLT/ELSIE were missing some of the features in the standard version because at that time we had not created Finite State Grammars for HAPI that included alternatives and key word spotting. Also, the verbal commands were not implemented in the test versions. However, in other respects the test versions performed well and were comparable to the earlier versions of QRSLT/ELSIE before the additional translation features and multiple language recognition were introduced.

The next step was to incorporate simultaneous English and Spanish input into HAPI QRSLT/ELSIE and attempt to improve overall speech recognition performance so that it would be superior to that provided by the IBM ICSS/VTAF recognizer.

6.8 Dual Language Implementation (QRSLT/ELSIE V.18)

The focus thus shifted to implementing recognition of both English and Spanish simultaneously using the HAPI interface. The remainder of this section describes how multiple-language recognition was performed by QRSLT/ELSIE using IBM VTAF, and how it would be done with Entropic's HAPI recognizer. The technical details of the dual-language implementation under HAPI are presented, along with test results comparing the strengths and weaknesses of both the IBM and HTK recognition systems.

One of our design goals in building QRSLT/ELSIE was to make the system sensitive to multiple languages simultaneously. This allows the system to be used in a conversational mode between a speaker of English and another language.

However, dual-language sensitivity does increase the difficulty of speech recognition, which is already a formidable task. If input can come from either of two languages, the recognizer faces the task of deciding which language is being spoken as well as what has been said. We can retain dual-language capability and remove the requirement of language identification by other methods, such as requiring the speakers to speak in turn or having the user press a button to choose which language will be used for a given utterance, but each of these workarounds compromises the conversational model that we envisioned for QRSLT/ELSIE.

6.8.1 Multiple Input Languages with IBM VTAF

One of the limitations of ICSS/VTAF was that the only language model it supported was English. Still, we were determined to experiment with multiple language recognition, so we created recognition contexts in Spanish and Mandarin by bootstrapping from the English language model; in other words, the Spanish and Mandarin words we wished to recognize were phonetically encoded, as nearly as possible, in English.

The resulting Spanish and Mandarin recognition contexts performed better than we had expected, although their performance was well below the level of their English equivalents.

6.8.2 A Multiple-Recognizer Algorithm for Supporting Multiple Input Languages

With Entropic's HTK/HAPI recognizer, we had a system with genuine English and Spanish language models. However, using such a system required a change in our algorithm for speech recognition. When all input for both Spanish and English was being processed by an English language recognizer, no adjustment to the internal recognition method was necessary when both languages were used at the same time. The decision as to which language had been spoken was made by looking at which recognition context supplied the successful result. Cases of an utterance in one language being mistaken for an utterance in the other language did occur, but they were rare.

With two separate language models, an input utterance must be processed by two separate recognizers. Then a decision has to be made as to which recognizer produced the correct result. In an ideal world the English utterances would all fail when fed to the Spanish recognizer and vice versa. But in reality, using two separate recognizers introduces the possibility that both recognizers will come up with a hypothetical recognition for a given utterance. How can we decide between them? As an interim solution we decided to choose the recognition with the highest "likelihood value" reported by the recognizer.

6.8.3 Modifying the Speech Recognition Object to Support Two Recognizers

As mentioned previously, QRSLT/ELSIE's speech recognition is encapsulated in a C++ object called SpeechRec. Performing speech recognition with two separate language models required modifications to this module.

Making this work required some changes to the SHAPI interface we had used with a single language. SHAPI includes an InitializeRecognizer() function in which the name of a configuration file is passed as an argument. At first we thought we could simply create two separate recognizer sessions like this:

```
m_pEngHappyStruct = InitializeRecognizer(HAPI_US_CFG_FILE);
m_pSpanHappyStruct = InitializeRecognizer(HAPI_SP_CFG_FILE);
```

However, this did not work. The problem is that the SHAPI InitializeRecognizer() function contained a call to the HAPI function hapiInitHAPI(), which apparently could only be called once per HAPI session, regardless of how many recognizer objects were created. In addition, the configuration file whose name was passed to InitializeRecognizer() and then to hapiInitHAPI() was the only configuration file that the system could use. The only way to implement a second recognizer with different language model parameters was to use multiple "Override" calls to change the parameters after creating the recognizer but before initializing it.

After the necessary modifications the prototype of InitializeRecognizer() looked like this:

```
SR_HAPI_Struct *InitializeRecognizer( char *DictFileName,
                char *HmmListFileName, char *MmfFileName,
                float GenBeam, float WordBeam);
```

Instead of initializing the recognizer with the name of a configuration file, a default configuration file was used and the InitializeRecognizer() function passed in parameter values that should be overridden. The parameters shown in the prototype specified the names of the files containing the dictionary, the list of phoneme models, and the language model, plus two floating point numbers to control the "beam width" of the search process. Other parameters could be added in the future as needed.

6.8.4 Testing Dual-Language Recognition

Once we had a test version of the SpeechRec object working, we incorporated it into a command-line program that would allow us to test large numbers of files with various recognition parameters. We already had such a program for testing recognition with the original ICSS/VTAF recognizer, so now we could test the two systems in parallel with the same input.

Then we put together a set of test recordings. We began with English recordings made at LSI of all 48 sentences in the Law Enforcement and Medical corpora that we began with as the first QRSLT/ELSIE finite state grammars. Then we added Spanish versions of the same 48 utterances by a Peruvian linguist, also done at LSI. Finally we added 10 "noise" recordings taken from a recent military tech expo in Massachusetts where QRSLT/ELSIE was being demonstrated. These noise recordings contained pops or clicks from handling the microphone, background noise, faint nearby conversations, and coughs.

The English and Spanish recordings were made at 16KHz and downsampled to 11.025KHz (using Entropic's ESPS) for the tests on the ICSS/VTAF recognizer. The noise recordings were done at 11.025KHz and upsampled to 16KHz for HTK/HAPI.

During the testing we distinguished between three types of errors:

1) **FAILED RECOGNITION** - The recognizer returned an error code indicating that it could not identify the input.
2) **INCORRECT RECOGNITION** - The recognizer thought it had made an identification, but the string returned was not what the utterance contained.
3) **WRONG LANGUAGE** - The recognizer made a correct recognition of the utterance in one language, but the recognition made in the other language had a higher likelihood value. This includes cases in which one recognizer correctly identified noise as noise, but the other recognizer came back with a text string.

6.8.5 Analysis of Test Results

When we were testing the recognizers separately and not using any "noise" recordings, as described in Section 6.7, we found that we could get perfect recognition for both languages if we made the GENBEAM and WORDBEAM parameters large enough. But when we ran the recognizers at the same time on the same recorded WAV files and also introduced noise files, as described above, we got very poor results. This prompted us to try many combinations of GENBEAM and WORDBEAM values for the two recognizers and led us to the following conclusions:

- The secret to getting good rejection of non-speech "utterances" is to keep both parameters, especially WORDBEAM, as small as possible.
- If WORDBEAM is kept small, the simple method of picking the recognizer that has the highest likelihood for a given utterance does a fairly good job of distinguishing between English and Spanish.

After extensive testing we found that we got the best results using a GENBEAM value of 200 and a WORDBEAM value of 55. The results were:

Recognized 97 of 106 (91.5%):
2 Wrong Language (English) errors
0 Wrong Language (Spanish) errors
4 Incorrect Recognition errors
3 Failed Recognition errors

In both cases the Wrong Language errors happened with noise recordings which the Spanish recognizer correctly identified as noise but the English recognizer recognized as "hello."

The IBM ICSS/VTAF recognizer had a very poor overall accuracy score for this same data set because of recognition failures on Spanish utterances:

Recognized 69 of 106 (65.1%)

0 Wrong Language (English) errors
0 Wrong Language (Spanish) errors
0 Incorrect Recognition errors
37 Failed Recognition errors

All English and noise recordings were recognized correctly, but only 23% of the Spanish recordings were recognized. This illustrates, once again, the unsuitability of using an English language recognition model for a native speaker of Spanish.

Testing by Entropic on the same data sets confirmed our results, as discussed in Section 8, Entropic's report of work performed on the QRSLT effort.

6.8.6 False Positives and Wrong-Language Recognition

In terms of recognition accuracy, the Entropic HTK/HAPI recognition system outperformed ICSS/VTAF in this test by a wide margin. However, the ICSS/VTAF system was superior in its rejection of "false positives." The Entropic HTK/HAPI recognizer was wrong 9 times out of 106, but 6 of those errors would have caused QRSLT/ELSIE to speak an erroneous output sentence. The IBM ICSS/VTAF recognizer made 37 recognition errors, but none of them would have caused erroneous output to be spoken.

This "false positives" measure is important in a system that may potentially be used in a noisy environment. A recognition error resulting in silent failure is far preferable to an error in which the system makes an incorrect recognition and speaks an inappropriate translation.

6.8.7 Issues Involving Processing Speed

Another problematic factor with the HTK/HAPI recognition system was its comparatively slow processing speed. Although the VTAF recognizer could process a spoken input rapidly enough that the total throughput time on a Pentium 133 or 150 notebook computer was at conversational speed, the HTK ASR operated at a substantially slower pace. This disadvantage was dramatically illustrated at a consortium meeting where LSI's system administrator spoke into two microphones attached to two notebooks, one with QRSLT/ELSIE using VTAF and the other with another copy of QRSLT/ELSIE using the HTK recognition system. Recognition with the latter system lagged significantly behind recognition with the VTAF software.

At this point, we requested that Entropic address the speed and wrong language recognition problems, which they were planning to do. The speed issue was to be solved by a later version of the alpha HTK software that was provided to us by Entropic in July of 1997, and work on fine tuning the accuracy of recognition with two recognizers running simultaneously was being performed by Entropic personnel at their Palo Alto facility.

6.9 Transition to the IBM Via Voice Recognizer (QRSLT/ELSIE V.2.1)

Much of LSI's development effort in the QRSLT/ELSIE project was focused on integrating the separate components of speech recognition, translation, and text-to-speech output into a coherent user interface. The user interface design developed by LSI, in its role as system integrator, was intended to modularize the recognition, translation and output functions as much as possible to reduce dependence between the modules. At the same time, the attempt was made to combine these three modules in a manner that would appear seamless to the user.

Since Entropic's HTK speech recognition system was not available until the first quarter of the second year of the project, , a beta version of IBM's ICSS/VTAF recognizer was used to process spoken input in English, Spanish, and Chinese. To increase the modularity mentioned above and pave the way for the eventual incorporation of Entropic's HTK, LSI chose to implement speech recognition from saved WAV-format files in a separate execution thread. The result is a program that is essentially recognizer-independent. When a beta version of the Entropic recognizer was made available, LSI was able to construct and demonstrate alternate versions of the QRSLT/ELSIE program using the IBM and Entropic recognizers, as mentioned in the preceding section.

Because the HTK alpha version had problems in achieving reliable dual-language speech recognition and ran at an unacceptably slow processing speed (see Sections 6.8.6 and 6.8.7), LSI developed a version of QRSLT/ELSIE based on IBM's new Via Voice recognizers for English and Spanish. For some time before this, IBM had been requesting that we discontinue using ICSS/VTAF, as it would soon become unsupported. We had believed that we would be able to convert to the HTK recognizer at some point, but the speed and accuracy problems were causing additional delay. In order to maintain acceptable levels of processing speed and accuracy in the QRSLT/ELSIE system while Entropic was attempting to solve these problems, we agreed to try the Via Voice recognizers in the version of QRSLT/ELSIE that was being demonstrated by Rome Laboratory in military applications and by LSI at trade shows, as well as being utilized experimentally by the Fresno County Sheriff's Department. Although the Via Voice English recognizer was a commercial product, we had only a beta version of the Via Voice Spanish recognizer.

However, since Entropic unexpectedly closed down their Palo Alto office in the last quarter of the project, a later version of the HTK software was never delivered to LSI.

Thus, the speech recognition component in the final version of the QRSLT/ELSIE system was driven by the IBM Via Voice recognizers for English and Spanish rather than the PC version of Entropic's HTK software.

7 THE SPEECH GENERATION COMPONENT

7.1 Background

In the QRSLT system, speech can be generated utilizing one of two available technologies: text-to-speech (TTS) or digital audio playback (DAP). TTS is comprised of a complex series of linguistic rules and routines that transform ASCII orthographic text into phonetics, which the synthesizer then uses to produce audible output. DAP is recorded actual speech; the speech signal is digitized, (usually) compressed, and then stored on disk. At playback time, the stored data is uncompressed, sent through a digital to audio converter, and then output.

Both technologies are made available to the user because both methods have advantages and disadvantages depending on the nature of the application. There are three advantages that DAP has over TTS: auditory quality, more cost-efficient processing power and less-involved extensibility to other languages. Since DAP is recorded real speech, it sounds exactly like the person who donated the speech. TTS synthesizers, which are produced by a machine, do not currently sound like natural human speech. In addition, due to the extra computational power needed for TTS, DAP is less costly in terms of processing power. Another advantage DAP has over TTS is that if there is a requirement for extensibility to other languages, using DAP requires only the recording of the outputs in that new language, although this is a labor-intensive process as noted below. TTS requires expensive re-engineering of the orthographics, phonology, phonetics and synthesizer parameters of the TTS synthesizer, i.e. building a new linguistic model for each language added.

The major advantage that TTS has over DAP is that with TTS, the user has at her disposal all the phonemes of the language to create an infinite, unplanned number of words and sentences, whereas in DAP, the output must be planned and recorded in advance. Also, in DAP, the process of digitizing and organizing speech samples for a reasonable application can be an extremely labor-intensive task. A second advantage of TTS is that its the data storage requirements are less than that of DAP, which must store an indeterminate number of words and sentences.

The nature of the application drives the choice of technology. For example, if the QRSLT device is to be used as an automated language trainer, DAP is superior to TTS since an important aspect of learning a second language is acquiring the ability to distinguish and reproduce the sounds of that language.

In general, DAP will be superior to TTS for applications where output quality is critical and the vocabulary is completely known in advance. For applications where the speech output is not completely determined in advance, TTS is the only solution for speech generation.

7.2 ETI's Text-to-Speech Technology

The TTS technology employed in the QRSLT system is ETI-Eloquence. ETI-Eloquence is a linguistically-sophisticated, multi-language and multi-voice text-to-speech system produced by Eloquent Technology, Inc. (ETI). Versions of the system are currently available for General American and British English, Castilian and Mexican Spanish, Parisian French, German, and Italian. ETI-Eloquence is fundamentally distinguished from other text-to-speech systems by its use of powerful and innovative linguistic models and accompanying software development tools that underlie the synthesis rules (Hertz 1988, 1990a, 1990b, 1991, 1997a, 1997b; Hertz and Huffman 1992; Hertz, Kadin, and Karplus 1985).

Unlike systems for limited-vocabulary applications, in which only a restricted set of utterances-defined in advance must be produced, an unrestricted text-to-speech (TTS) system must be able to produce intelligible speech for any input text in the language in question. To accomplish this task, TTS systems generally have two main components: a *text module* and a *speech module.* The text module contains the algorithms, or rules, that analyze the input text into linguistic units like sentences, words, and phonemes, and assign relevant features to these units based on contextual information. The speech module uses the information produced by the text module to assign the actual acoustic values which are sent to the synthesizer to produce the speech output.

Text-to-speech products differ in the strategies they use to parse the input text into these linguistic units. ETI-Eloquence uses a novel and powerful approach developed through over 20 years of research by Dr. Susan Hertz and her associates at both Cornell University and ETI. This approach centers around a unique, multi-tiered utterance representation called a *delta,* in which all of the linguistic units necessary for high-quality speech generation (sentences, intonational phrases, words, morphemes, syllables, and phones) are explicitly represented. The following, for example, is a fragment of the delta produced by ETI-Eloquence for the word *untied.*

```
text:    |u |n   |t|i  |e|d  |
word:    |wrd                |
syllable: |stress2 |stress1    |
morph:   |prefix  |root  |suffix|
phone:   |H |n   |t|a|y |d    |
```

Each labeled horizontal line, called a *stream,* represents a structural level of the utterance. The streams are defined by the developer in the rule program, so they can be modified for the needs of a particular language. For example, in English a morph stream is necessary for predicting the phones for the utterance: compare, for example, the pronunciation of the letters *ed* in *naked*, in which these letters do not constitute a suffix, with the same letters in *baked*, in which these letters do constitute a suffix. In Spanish, on the other hand, the correspondence between spelling and phones is quite direct, so the same kind of morphological analysis is not required to predict the phones.

Each stream of the delta contains a sequence of tokens, such as n, t, etc. in the phone

stream. The vertical bars, called *sync marks*, coordinate the tokens across streams. The tokens in each stream also have various features *(fields)* associated with them, which the rules in ETI-Eloquence refer to in order to make appropriate linguistic generalizations. For example, tokens in the phone stream are marked for place of articulation, manner of articulation, voicing, etc. The parsing algorithms that make up the ETI-Eloquence text module are formulated in ETI's special Delta programming language, which was developed over a ten-year period specifically for the straightforward expression and testing of rules that operate on multi-tiered deltas (Hertz, Kadin, and Karplus, 1985; Hertz, 1990b). Typically these rules test the delta for particular patterns and manipulate it accordingly.

The speech module uses the information generated by the text module to produce speech output. A speech module must generate the appropriate acoustic values for each individual speech sound of the utterance, as well as its overall prosody–i.e. its timing and intonation. Acoustic values for individual speech sounds depend on properties of the linguistic structure such as those exemplified in the delta above. Prosodic rules make crucial reference to the field values of a variety of streams, including the word stream, which carries information about the relative stress patterns of individual words within the phrase and their accentual characteristics, and the intonational phrase stream, which stores information about properties of the phrase as a whole. A sample delta fragment showing selected fields and streams in the representation of the sentence *"John came," he said.* is shown below:

```
sentence:     | |sentence                          | |
inton_phr:    | |phrase1          |     |phrase2        | |
type:         | |comma            |     |period         | |
nuc_accent:   | |yes              |     |no             | |
phrase_tone:  | |low              |     |low            | |
boundary_tone:| |low              |     |low            | |
pause_length: | |150              |     |400            | |
text:         | "|j|o|h|n|' '|c|a|m|e|,|"|' '|h|e |' '|s|a|i|d|.|
word:         | |1    | |2    |     |3   | |4    | |
category:     | |undef | |undef |     |pro | |undef | |
function:     | |content| |content|     |funct| |content| |
stress_level: | |0    | |2    |     |0   | |1    | |
accent:       | |high | |high |     |high | |high | |
```

Unlike *concatenative* synthesis systems, which piece together speech fragments that have been extracted from natural speech, ETI's *rule-based* system generates all the necessary acoustic values by rule from abstract linguistic representations like those illustrated above (see Hertz 1997b for a contrastive discussion of these two approaches to speech synthesis). In both the text and the speech modules, these rules are grouped into universal, language-specific/dialect-universal, and dialect-specific components. The relative size of each of these components varies from one portion of the program to another. The speech module contains a large language-universal component, made possible by innovative phonetic models developed by Hertz and her collaborators (see

e.g. Hertz 1991, Hertz and Huffman 1992 on the phone-and-transition model), and smaller language- and dialect-specific ones.

7.3 System Status for the First Year of Development

7.3.1 Work on American English and Mexican Spanish

During the first year of the QRSLT/ELSIE development, ETI created a new integrated TTS program for American English and Mexican Spanish as a Windows Dynamic Link Library (DLL) for use in QRSLT/ELSIE. This program is an improvement from the previous version in two important ways: it generates higher quality speech, particularly for Mexican Spanish, and it also incorporates universal, and thus, a smaller set of restructured rules. The universal components presently comprise about half of the current Mexican Spanish rules. The DLL for English and Spanish together is about 1.34 megabytes in size. This DLL is smaller than our previous DLL for English and Spanish, even though there are now substantially more Spanish-specific speech rules.

After a Spanish model speaker was selected in a rigorous manner, a number of necessarily reiterative steps described below were taken to create this DLL. The Spanish model speaker was selected in the following way: first, the preliminary phrase list ("QDC development corpus") from two Mexican Spanish speakers was recorded, as well as several narrative texts and phrases read by the same two speakers. Then, a preliminary evaluation of both speakers was performed to determine how well they might serve as models for the Mexican Spanish speech component of the translator. The evaluation was carried out by examining spectrograms of the speakers, and, with Entropic's and LSI's help, by asking a few Latin American Spanish speakers to listen to the speech and answer the following kinds of questions:

What is your general reaction to this speaker's voice?
Can you tell where he/she is from? If so, where?
Does the speaker sound particularly educated or uneducated, or can you not tell?
Does the speaker sound pleasant and/or friendly?
Is the speaker clear and easy to understand?
Is there anything remarkable or unusual about the voice?
Can you judge the physical size of the speaker from his/her voice? If so, please describe how you think he/she looks.
Can you judge the age of the speaker from his/her voice?
If so, how old do you think he/she is?
What is your general image of this speaker?

The listeners preferred the first speaker to the second for a number of reasons. They correctly identified the first speaker as being from Mexico or Central America, though some thought Colombia was a possibility, too. They thought the speaker to be educated, in his thirties, and generally found his voice pleasant and clear. It appeared that this speaker had been raised in a primarily Spanish-speaking environment, and was accustomed to reading Spanish, whereas the second speaker was evaluated as being U.S.-

born, and unaccustomed to reading Spanish. In addition to being generally well liked by the listeners, the speech of the first speaker produced clear spectrograms, thus was especially appropriate for the speech analysis task of the project.

After the Spanish model speaker was decided upon, his voice was periodically recorded and digitized using the QDC development corpus. While the speech database was being expanded in this way, acoustic analyses and literature research of the sounds not having English correlates were performed; these phonemes included the bilabial fricatives, the trilled and flapped [r], and the velar fricative. As more acoustic information about the Spanish speech of the model speaker was gathered, the multi-tiered delta rules (as described above in ETI's TTS Technology section) were restructured to reflect the new linguistic information, which then allowed the Spanish consonants listed above to be interactively synthesized and evaluated. Since acoustic information such as pitch and duration is gleaned crucially from spectrographic modelling, ways were explored to automate this method which involves a lot of time-consuming manual segmentation of phonemes. Also, the advantages and disadvantages of various approaches for prestoring utterance information such as pitch and duration were evaluated; other spectral values would be filled in by rule. (Prestored utterances are compact, parametric representations of specific utterances, which are derived from spectrograms.) But since these prestoring approaches also involve time-consuming manual segmentations of spectrograms, they would be worth incorporating only if the rules themselves did not synthesize good speech. Therefore, the focus in the first year of the project was on improving the rules that synthesize American English and Mexican Spanish. Another reason for concentrating on developing the rules rather than on, e.g., modelling, is that the present technology now allows for successful, easy integration of the rules of several languages into one DLL to be employed in the second prototype of QRSLT/ELSIE.

In integrating both English and Spanish rules into one DLL, a total reorganization of the speech rules was conducted. Universal components in both languages were factored out and language-specific rules were developed for each language. For example, voicing amplitude values for a syllable nucleus are universally positioned at the beginning and end of the nucleus. Before the reorganization, the amplitude rules for *each* language explicitly inserted the appropriate amplitude values at the beginning and end of the nucleus. As the reorganized rules now stand, the amplitude rules for each language simply specify the appropriate amplitude values, and a universal procedure inserts these values at the appropriate points in the delta utterance representation. The result of factoring out universal rules is that the rules are more consistent across languages, they are faster to develop, create a shorter learning curve for new developers, smaller programs for each language, faster rule execution, and easier integration of several languages into a single program.

For future work, aside from improving the rules as stated above, special annotations for TTS for better prosody will be further developed. The quality of many sentences can be improved significantly by strategically marking the input text with special intonation annotations (e.g. to emphasize certain words) that are understood by the ETI-Eloquence TTS rule program, and by entering words not pronounced correctly in terms of phonetic

symbols, rather than orthography. In addition, the function that allows application developers to speed up or slow down synthesized speech will be implemented within the new rule framework.

7.4 The Second Year of Development

7.4.1 Work on Mandarin Chinese

The major achievements during the second year of this reporting period are that a Chinese synthesizer was developed, the language universal component was enhanced, and the American English and Mexican Spanish synthesizers were further improved and refined. The following is a summary of the work done in each of these areas.

Chinese Synthesizer Development: In preparation for the development of the Chinese synthesizer, some preliminary recordings were made with potential model speakers for evaluation. The recordings were based on our designed structured data set. Then, preliminary modules were configured for the Pinyin-to-speech component, that would be developed using the Delta system. (Pinyin is the Romanized version of the Chinese characters.) In particular, preliminary Delta modules were written for interpreting the Pinyin input script, and for producing the appropriate linguistic structure, i.e. phonetic symbols, associated features, tone values, and so forth, based upon it. The linguistic structure is depicted in the form of a multi-tiered delta representation, with words, syllables, tones, phonemes, etc. on separate tiers, or 'streams'. In addition, a preliminary speech module was written which is sensitive to the linguistic information in the delta and produces speech output based upon it. The Chinese synthesis rules were developed by using a top-down approach; i.e. an appropriate global structure was created for the rules, which were then refined with specific procedures in separate iterations throughout the rules on the basis of further data analysis. For example, in the text module, a preliminary set of tone sandhi rules (i.e. the way tones are affected by neighboring tones) was posited which were then simplified by making tone a property of the syllable, rather than a separate unit in its own delta stream; the glide insertion rules were improved as well. As another example, preliminary duration rules, based upon comparable phones in other languages, were posited in the acoustic module, which was all that was needed to generate spectral values for the phones. These values were then refined as appropriate for Chinese.

Improvements to the Chinese synthesis rules for spectral values and durations of individual phonemes were made by extensive measurement of acoustic values for vowels and consonants produced by the potential model speakers. Preliminary procedures were written to produce the pitch values for each syllable based on the tonal specifications of the syllables, which are derived from the Pinyin input. In addition, a strategic data set was written for obtaining duration measurements. A native Chinese speaker was trained in the Delta system in order to work on the Chinese pitch and intonation rules.

After preliminary working Chinese modules were set into place, one model speaker was selected from the potential Chinese speakers and evaluated on the basis of voice quality and formant patterns. This speaker served as the model speaker for the remainder of the rule development. Rules for tone sandhi and for predicting specific pitch values from the

syllable tones were developed. To improve duration rules, data sets were first developed and recordings then were made for nucleus, consonant, and transition durations. Measurements were then made in order to formulate preliminary rules for nucleus durations. Using the measurements from the newly selected model speaker's recordings, vowel formant rules were revised.

As the Chinese synthesis rules were being developed and enhanced further, the TRP data was supplemented by new Chinese corpora collected from news articles and other texts. These texts, which were written with Chinese characters, were converted into Pinyin. These new texts are being used to improve the generality of the rules.

Text processing rules to handle digits and years were added. More speech data was gathered. Test versions of the system in a DLL for Windows-based PC's were created for evaluation. (The rules in DLL format differ from the rules as they are being developed with the Delta system.) The Chinese synthesizer is ready for release in the next upgrade version as it is now highly intelligible and more natural-sounding. Future plans are to continuously improve the rules and unite them with a text module that can take Chinese characters as input, rather than only Pinyin. The system will also be made compliant with the standard speech application programming interface, SAPI, thereby making the product useable in other types of speech applications than just the particular goal of this QRSLT project.

7.4.2 Language Universal Component Enhancements

Most of the work in this component up to this point has concentrated on the speech module. Currently, the text module is being developed which includes general text processing procedures which divide the text into words, phrases, and sentences, and procedures for intonational analysis which will determine pitch accents, word stress, and so forth. The rules in this component will expedite future development of rules for Chinese and Mexican Spanish pitch patterns and other English rules as well.

7.4.3 American English Synthesizer

Extensive testing and refining of the American English rules were continued throughout the project. These refinements included [tw] and [kr] syllable onsets, word-final [t] before word-initial labial stops, [sp] and [st] clusters followed by sonorants at syllable onsets, amplitudes of [m] and [y], amplitude of [k] aspiration before [y], durations of selected syllable nuclei (e.g. the final 'er' nucleus of words like 'better' and 'favor'), formant transition durations between selected segments, and certain vowel formant and amplitude values. Problems in specific voices, such as the [g] bursts in the female voice, were also addressed. As the English TTS system was being improved at various levels, such as speech quality, robustness, documentation, a new integrated DLL was created for English and Spanish. As the rules were improved further, better rules were also written for coarticulation (adjustment of acoustic values based on context). Work on voice quality included experimentation with various kinds of manipulations to the acoustic parameter values generated by the rules to determine how we could improve the overall naturalness of the voice quality that is produced.

As the Spanish and Chinese synthesizers were being developed, the English TTS system was being continuously upgraded with further refinements in the text-to-phoneme module, text normalization module, and intonation module by correcting the rules for relative stress levels of words in the sentence. The text normalizer converts numerical expressions into letter sequences and expands abbreviations, acronyms and the like, for interpretation by the letter-to-sound rules. New rules were written to handle specific numerical expressions such as dates, telephone numbers, times, monetary expressions , and zip codes. Existing English normalization rules were reworked into a universal structure that serves as the basis for the text-processing of all languages that use the Roman alphabet. By incorporating rules into the universal component as much as possible, rule repetition in the text module is avoided, which simplifies maintenance of the rules, as well as reducing the size of the multi-language systems.

The text intonation rules divide sentences into intonational phrases and predict the pitch accent patterns of the words in the phrase. Rules were written which improve the prediction of which words in a phrase get the pitch accents. For example, in the sentence "I want the one over there", the word "one" is unaccented, whereas "one" in the context of the following sentence "I only have one dog", gets the pitch accent. The rules can now predict when "one" is accented and when it remains unaccented. Work was also done on recognizing and generating appropriate pitch accent patterns for noun compounds in which the first word is more stressed than the second, such as "beer can" vs. "tin can", "Elm Street" vs. "Elm Road", and "White House" vs. "white house". New rules for creating accurate postnuclear intonation contours and intonation contours for phrases lacking nuclear accents were also written. Work on prosody also included a variety of improvements to function word and consonant durations as well as to intonation for phrases lacking nuclear-accented words, post-nuclear intonation contours, and exclamatory sentences. Segmentals, with an emphasis on nasals, were improved by adjusting formant values and other acoustic parameters. Amplitude rules were added which lowered amplitudes on unstressed syllables.

7.4.4 Mexican Spanish Synthesizer

A variety of improvements were made in the Mexican Spanish synthesizer. The procedure for assigning durations to syllable nuclei was substantially improved. New procedures were added for assigning formant values to classes of sounds such as dentals, velars, labials, and trills. In addition, rules for [l] in a variety for contexts were improved and nasal consonants were improved b adjusting their bandwidths in appropriate contexts. The quality of [y] in particular contexts was also improved. In addition, selected formant transition durations were improved as well as the formant patterns of the five vowels of Spanish. In an ongoing effort of testing the system for robustness, a variety of bugs were uncovered and fixed. A number of Spanish jailbook utterances were also digitized for QRSLT/ELSIE prototype.

Improvements to the Spanish TTS system continued to be made at all levels: speech quality, robustness, and documentation, as well as work on intelligibility and naturalness testing with native Spanish speakers. A newly integrated DLL was also created for Spanish and English. Spectrograms of selected Spanish utterances from the TRP list were

modelled. (See First Year of Development section above for discussion on modelling.) Fully-modelled versions were compared with hybrid versions, in which the spectral values for the segmentals were generated by rule and the pitch and durations extracted from the spectrograms.

In the speech module, rules were improved for formant values of palatals, for vowel coarticulation (i.e. formant changes based on surrounding consonants), for formant transitions after fricatives, and for selected aspects of approximants and trills. The duration rules for nuclear words (i.e. for the most accented word in each phrase) were improved as were rules for selected spectral values of phones.

In the text module, normalization rules were improved for numerics and syntactic parsing rules were improved to better predict intonation patterns. Refinements were also made to the rules that predict pauses and to the rules that produce acoustic values for individual speech segments. Formant rules were restructured to centralize rules that are common to groups of segments. Rules were also profiled to determine performance bottlenecks, and improved some of the code accordingly.

Modifications were made to the text-to-phoneme rules for improved accuracy; rule profiling, i.e. tracking the amount of run-time used by individual procedures in the program, as an initial step in improving the over-all efficiency was conducted as well.

7.4.5 General Synthesizer Technology Development

As work progressed on the development of the Chinese, Spanish and English synthesizers, and the language-universal components were enhanced, other types of technological developments were also achieved. Language-swapping is one of these. English and Spanish had been integrated into a single DLL, but an alternative strategy for integrating multiple languages that does not require all the languages to be available in a single program at once was developed. In the new language-swapping strategy, when a language is invoked via a textual annotation or a call to the text-to-speech API, the DLL for that language is swapped into memory, and the DLL for the previous language, if any, is swapped out. This swapping can occur at sentence boundaries, and is transparent to the user. This strategy will keep memory requirements at a minimum as more and more languages become available (especially once the universal components are factored out in the future into a separate shared DLL), and will facilitate distribution and maintenance of the product for different combinations of languages. Another side-line development is that the on-line help file for the text-to-speech part of the product has been enhanced. Among other things, the help file explains how to customize the speech, for example, to create different voice characteristics or intonational effects. In addition, a number of improvements to the Delta system (the main rule development tool) and the Delta preprocessor have been made. With the preprocessor, 'tags' can now be placed into the Delta programs (i.e. the synthesis rules) much more expediently in order to conditionally compile code for different languages. For example, any code between the tags "::-english spanish" and "::-english spanish end" will be compiled and executed for those languages, but not for, say, Chinese. Similarly, any code between "::!english" and "::-english end" will be compiled for all languages except English.

8 ENTROPIC SPEECH AND LANGUAGE RECOGNITION EXPERIMENTS

Three Entropic speech recognizers were integrated and evaluated for pairwise language identification and utterance recognition accuracy under varying parameter values. Spanish, Mandarin Chinese and English recognizers were used.

Initial tests, varying only the values of the parameters GENBEAM and WORDBEAM, were unable to provide 100% language identification accuracy, so two other methods of language identification were applied, one using confidence measures and the other using acoustic scores.

For choosing between English and Spanish, confidence measures yielded a more accurate indication of which language was spoken. For each input utterance, Entropic speech recognizers provide a score from 0 to 1, the confidence measure, which indicates how confident the recognizer is that the hypothesized string is the correct one. Each sentence was fed into both the English recognizer and the Spanish recognizer and the confidence scores from the recognizers were then compared. The language of the higher scoring recognizer was assumed to be the language of the input. On average, the English recognizer yielded higher confidence scores than the Spanish recognizer, so scores were normalized by subtracting 0.09 from the English confidence score. The textual output of the higher scoring recognizer was provided as the recognized utterance.

For choosing between English and Chinese, acoustic scores produced more accurate results.

9 REFERENCES

Arnaiz, A.R., R.S. Belvin, N. Li, S.H. Litenatsky, C.A. Montgomery, B.G. Stalls, and R.E. Stumberger (in press) *Machine-Aided Voice Translation (MAVT): Advanced Development Mode*, Rome Laboratory/IRAA, LSI 97-02, Contract No. F31602-93-C-0098.

Hertz, S. R. (1988) "Delta: flexible solutions to tough problems in speech synthesis by rule," *Proceedings of Speech Tech-88*, Media Dimensions, N.Y.

Hertz, S. R. (1990a) "A modular approach to multi-dialect and multi-language speech synthesis using the Delta System," *Proceedings of the Workshop on Speech Synthesis,* European Speech Communication Association, Autrans, France, 225-228.

Hertz, S. R. (1990b) "The Delta programming language: an integrated approach to non-linear phonology, phonetics, and speech synthesi,s" *Papers in Laboratory Phonology I: Between the Grammar and the Physics of Speech*, J. Kingston and M. Beckman (eds.), Cambridge University Press.

Hertz, S. R. (1991) "Streams, phones, and transitions: toward a phonological and phonetic model of formant timing," *Journal of Phonetics* 19, *Special Issue on Speech Synthesis and Phonetics*, edited by R. Carlson.

Hertz, S. R. (1997) "The technology of text to speech," *Speech Technology*, April/May , CI Publishing, 18-21.

Hertz, S. R. and M.K. Huffman (1992) "A nucleus-based timing model applied to multi-dialect speech synthesis by rule," *Proceedings of the International Conference on Spoken Language Processing,* Vol. 2, 1171-1174.

Hertz, S. R., J. Kadin, and K. Karplus (1985) "The Delta rule development system for speech synthesis from text," *Proceedings of the IEEE Special Issue on Man-Machine Speech Communication*, 1589-1601.

Montgomery, C., B.G. Stalls, R.E. Stumberger, N. Li, R.S. Belvin, A.R. Arnaiz, P. Shinn, A. DeCesare, and R. Farmer (1996) *Machine-aided Voice Translation (MAVT)*, Rome Laboratory/IRAA, RL-TR-95-265.

Montgomery, C., B.G. Stalls, R.E. Stumberger, N. Li, S. Walter, R. Belvin, and A. Arnaiz (1993) "Machine-Aided Voice Translation," Information Management Collection Processing & Distribution, Dual Use Technologies & Application Conference, *IEEE*, 96-101.

Montgomery, C.A., B.G. Stalls, R.E. Stumberger, N. Li, R.S. Belvin, A.R. Arnaiz, and S.H. Litenatsky, (1995) "The machine-aided voice translation (MAVT) system," *AVIOS '95 Proceedings*, 101-10.

Stalls, B.G., R.S. Belvin, A.R. Arnaiz, C.A. Montgomery, N. Li, R.E. Stumberger, and S.H. Litenatsky (1994) "An adaptation of lexical conceptual structure to multilingual processing," *Technology Partnerships for Crossing the Language Barrier (Proceedings of the First Conference of the Association for Machine Translation in the Americas)*, 106-13.

APPENDIX A. Sample of the Law Enforcement Dialog Corpus

18. viernes *Friday*
19. sábado *Saturday*

4 Some Expressions of Time and Direction

4.1 Time — *Hora*

1. segundo *second*
2. minuto *minute*
3. hora *hour*
4. día *day*
5. semana *week*
6. mes *month*
7. año *year*

4.2 Time of day — *Hora del día*

1. mañana *morning*
2. mediodía *noon*
3. tarde *afternoon/evening*
4. noche *evening/night*
5. ayer *yesterday*
6. hoy *today*
7. mañana *tomorrow*
8. esta noche *tonight*

4.3 Directions — *Direcciones*

1. norte *north*
2. sur *south*
3. este *east*
4. oeste *west*
5. izquierda *left*
6. derecha *right*
7. derecho/de frente/recto *straight*

5 Predominant Spanish Colors

1. rojo/colorado *red*
2. blanco *white*
3. amarillo *yellow*
4. azul *blue*
5. verde *green*
6. anaranjado/naranja *orange*
7. negro *black*
8. morado *purple*
9. gris *grey*
10. rosa/rosado *pink*
11. café/moreno/marrón *brown*
12. castaño *light brown/hazel*
13. claro *light color/clear*
14. oscuro *dark color/obscure*
15. dorado/de oro *golden*
16. plateado/de plata *silver*
17. rubio *blonde*

6 Clothing

1. ¿Qué lleva? *What are you wearing?*
2. chaqueta *jacket*
3. pantalones *pants*
4. camisa *shirt*
5. blusa *blouse*
6. corbata *tie*
7. vestido *dress*
8. falda *skirt*
9. traje *suit*
10. calcetines *socks*
11. zapatos *shoes*

18.	váyase allá	*go over there*
19.	váyase	*go away*
20.	quédese afuera	*stay outside*
21.	dígame pronto	*tell me quickly*
22.	dígame la verdad	*tell me the truth*
23.	deme la información	*give me the information*
24.	deme su licencia	*give me your license*
25.	firme aquí	*sign your name here*
26.	deletree(me) su nombre	*spell your name*

26.4 Vehicle Commands

1.	¡Apágue el motor!	*Stop the engine!*
2.	¡Súbase al carro!	*Get in the car!*
3.	¡Quédese en el carro!	*Stay in the car!*
4.	¡Bájese del carro!	*Get out of the car!*
5.	¡No mueva el carro!	*Don't move the car!*

26.5 Custody Commands

1.	¡Levante las manos!	*Raise your hands!*
2.	¡Ponga las manos atrás/ detrás de la espalda!	*Put your hands behind your back!*
3.	¡No se mueva!	*Don't move!*

26.6 Phrases with Soltar (To Release)

1.	¡Suelte la navaja!	*Release the knife!*
2.	¡Suelte la pistola!	*Release the pistol!*
3.	¡Suéltela!	*Release it/her!*
4.	¡Suelte el arma!	*Drop the weapon!*
5.	¡Suelte las llaves!	*Drop the keys!*

26.7 Kneeling Search

1. ¡Manos arriba!
   Hands up!

2. ¡Abra/separe los dedos!
   Spread your fingers!

3. ¡(Ahora,) voltéese despacio!
   (Now,) turn around slowly!

4. ¡Despacio, bájese a sus rodillas!
   Slowly, get down on your knees![1]

5. ¡Cruce los dedos de las manos detrás de la cabeza!
   Interlace (cross) the fingers of both hands behind your head!

6. ¡Junte las rodillas!
   Put your knees together!

7. ¡No se mueva!
   Don't move!

26.8 Standing/Cursory Search

1. ¡Manos arriba!/¡Levante las manos!
   Get your hands up!

2. ¡Abra/separe los dedos!
   Spread your fingers!

3. ¡(Ahora,) voltéese despacio y cruce los dedos de ambas manos detrás de la cabeza!
   (Now,) turn around slowly and interlace (cross) the fingers of both hands behind your head!

4. ¡Separe los pies/piernas!
   Spread your feet/legs!

5. ¡No se mueva!
   Don't move

27 Additional Information Phrases

1. Esta forma le indica que usted ha sido arrestado/a
   This form indicates to you that you have been arrested

2. Usted tiene que/debe de tener identificación
   You have to have identification

[1] A better translation would be: *¡Lentamente, arrodíllese!*

APPENDIX B. Samples from the Mandarin Chinese Dialog Corpus

Initial Dialogs: English-to-Mandarin Medical Sentences with Pinyin Translations

<sentence> ::= Are you injured? ^ Ni3 shou4shang1 le ma? .

<sentence> ::= Does your chest hurt? ^
Ni3 xiong1kou3 teng2 ma? .

<sentence> ::= Where does it hurt? Show me ^
Ni3 shen2mo di4fang1 teng2? Zhi3 gei3 wo3 kan4 .

<sentence> ::= You are injured, please do not move! ^
Ni3 shou4shang1 le, qing3 bie2 dong4! .

<sentence> ::= Are you ill? ^
Ni3 sheng1bing4 le ma? .

<sentence> ::= Are you a diabetic? ^
Ni3 you3 tang2niao4bing4 ma? .

<sentence> ::= Do you have heart trouble? ^
Ni3 you3 xin1zang4bing4 ma? .

<sentence> ::= How do you feel? ^
Ni3 gan3jue2 zen3moyang4? .

<sentence> ::= Are you taking medication? ^
Ni3 zhe4 duan4 shi2jian1 zai4 chi1 shen2mo yao4 ma? .

<sentence> ::= Where is your medicine? ^
Ni3 de yao4 zai4 shen2mo di4fang1? .

<sentence> ::= You need medical care ^
Ni3 xu1yao4 zhi4liao2 .

<sentence> ::= Do you want a doctor? ^
Ni3 xu1yao4 kan4 yi1sheng1 ma .

<sentence> ::= Do you want an ambulance? ^
Ni3 xu1yao4 jiu4hu4che1 ma? .

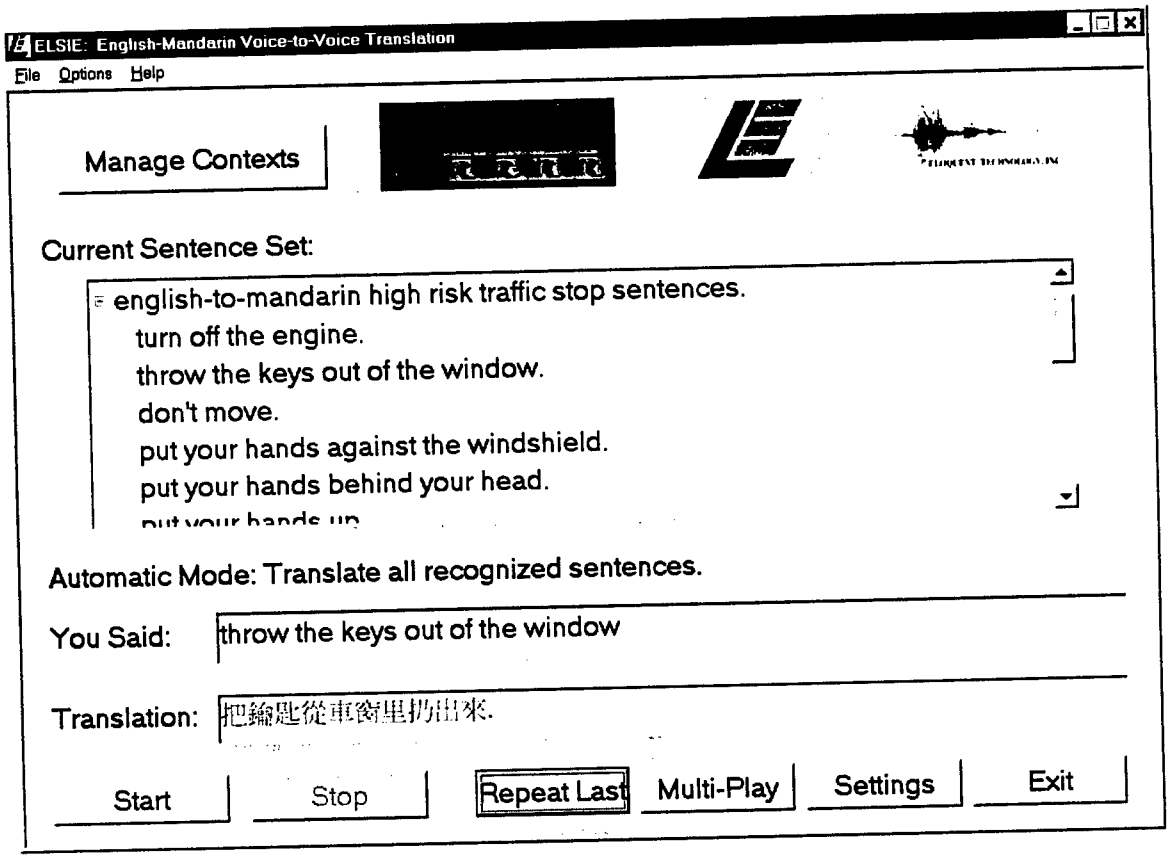
<sentence> ::= You should see a doctor ^
Ni3 ying1gai1 kan4 yi1sheng1 .

<sentence> ::= Do you want to go to the hospital? ^
Ni3 xiang3 qu4 yi1yuan4 ma .

<sentence> ::= You have to go to the hospital ^
Ni3 bi4xu1 dao4 yi1yuan4 qu4 .

<sentence> ::= Where is your medical card? ^
Ni3 de yi1liao2 bao3xian3ka3 zai4 shen2mo di4fang1? .

Sample from High Risk Traffic Stop Dialog Showing Translation into Traditional Chinese Characters



APPENDIX C. Samples of the Fresno County Dialog Corpus

Fresno Country Sheriff's Department Booking Questions

What is your nationality?
How much do you weigh?
How tall are you?
What color are your eyes?
What color is your hair?
What is your date of birth?
What year were you born?
How old are you?
What is the name of the city where you were born?
What state were you born in?
What country were you born in.
Do you have any tattoos?
Do you have any scars?
Do you have any birthmarks?
What is your social security number?
Do you have a driver's license?
What is your driver's license number?
What state issued your driver's license?
Do you have an identification card issued by the state of California?

Fresno County Sheriff's Department Medical Screening Questions

How long have you been on insulin?
Have you ever had a reaction to insulin?
Do you ever miss any doses of insulin?
Can you give us a urine sample?
When is the last time you saw your doctor?
Have you ever been on high blood pressure medicine?
Have you ever been told you have high blood pressure?
Does anyone in your family have high blood pressure?
If you have stopped taking your blood pressure medicine, when and why?
Have you ever been seen in an emergency room because of high blood pressure?
Have you ever had a heart attack?
Have you ever had a stroke?

APPENDIX D. Korean Dialog Corpus Sample and Development Summary for the Global Patriot Exercise

Stop!
정지!
Stop or I'll shoot!
멈추세요 아니면 총을 쏘겠습니다.
Do not move!
움직이지 마세요!
Don't move!
움직이지 마세요!
Drop your weapons!
무기를 떠러트리세요!
Don't shoot!
쏘지 마세요!
Hands above your head!
손을 머리위에 올려 놓으세요!
Hands over your head!
손을 머리위에 올려 놓으세요!
Hands up!
손을 드세요!
Put your hands on the wall.
손을 벽에 대세요.
Place your hands on the wall.
손을 벽에 대세요.
Surrender!
항복 하세요!
Do not resist!
저항하지 마십시요!
Don't resist!
저항하지 마십시요!
You will not be harmed.
해를 업히지 않겠습니다.
We will not harm you.
해를 업히지 않겠습니다.
Put your weapon down!
무기를 내려놓으세요!
Are you carrying a weapon?
무기를 가지고 계십니까?
You are a prisoner.
당신은 죄수입니다.
We must search you.
당신을 수색하겠습니다.
We have to search you.
당신을 수색하겠습니다.
We need to search you.
당신을 수색하겠습니다.
Turn around!
뒤로 돌으세요!
Lie face down!
엎드려 누으세요!

English-to-Korean Medical Sentences

Have you been wounded?
다치셨습니까?
Are you wounded?
다치셨습니까?
Are you injured?
다치셨습니까?
Are you hurt?
다치셨습니까?
Where are you injured?
어디를 다치셨습니까?
Show me where you are wounded.
어디를 다치셨는지 보여 주세요.
Show me where you are hurt.
어디를 다치셨는지 보여 주세요.
Show me where you are injured.
어디를 다치셨는지 보여 주세요.
Are you sick?
아프십니까
Are you ill?
아프십니까
Do you need medicine?
약이 필요합니까?
Do you need medication?
약이 필요합니까?
Do you need any medical attention?
의료 치료가 필요 합니까?
Do you need medical attention?
의료 치료가 필요 합니까?
Do you require any medical attention?
의료 치료를 필요로 합니까?
Do you require medical attention?
의료 치료를 필요로 합니까?
Do you need food?
음식을 원합니까?
Do you need water?
물이 피료합니까?
We have food.
음식이 있습니다.
We have water.
물이 있습니다.
Boil this water!
이 물을 끓으세요!
Drink this!
이 물을 마시세요!
Where is the doctor?
의사가 어디에 있습니까?
Where's the doctor?
의사가 어디에 있습니까?
He is there.
저기에 있습니다.

Development of a Korean Language Capability for QRSLT/ELSIE

A very limited capability for one-way translation into Korean was developed during the 7th quarter of the QRSLT project to demonstrate the feasibility of adding other Asian languages and character sets to the system. For the Global Patriot exercise, this capability was significantly enhanced and extended, and a limited Korean recognizer was developed to provide two-way translation from English-to-Korean and Korean-to-English. The development of this new capability involved efforts in five areas:
1) collection of Korean data; 2) construction of a path through the normal translation process; 3) representation and display of Korean sentences in both Hangul and romanized characters; 4) building of a limited Korean speech recognizer; 5) recording of Korean phrases to provide for the generation of output speech.

1) Korean data collection

Since the addition of the Korean capability was intended to serve as a technology demonstration for the Global Patriot exercise, the sentence sets for which Korean translations and responses were developed were mainly derived from the command and control cards and other materials produced by DLI for use by military personnel in special operations. The first set of sentences is more oriented toward a combat situation. Figure 1. shows a few sentences from this set.

The second set is more medically oriented, and might be used for handling medical problems of prisoners of war, or medical needs of the civilian populace in a peace-keeping operation (see Figure 2.).

2) Translation path

In the previous version of QRSLT/ELSIE that included Korean, the system had only a primitive English-to-Korean translation capability, which was added to demonstrate the feasibility of handling additional Asian languages and scripts. The translation strategy utilized in that version was essentially a sentence-to-sentence table look up.

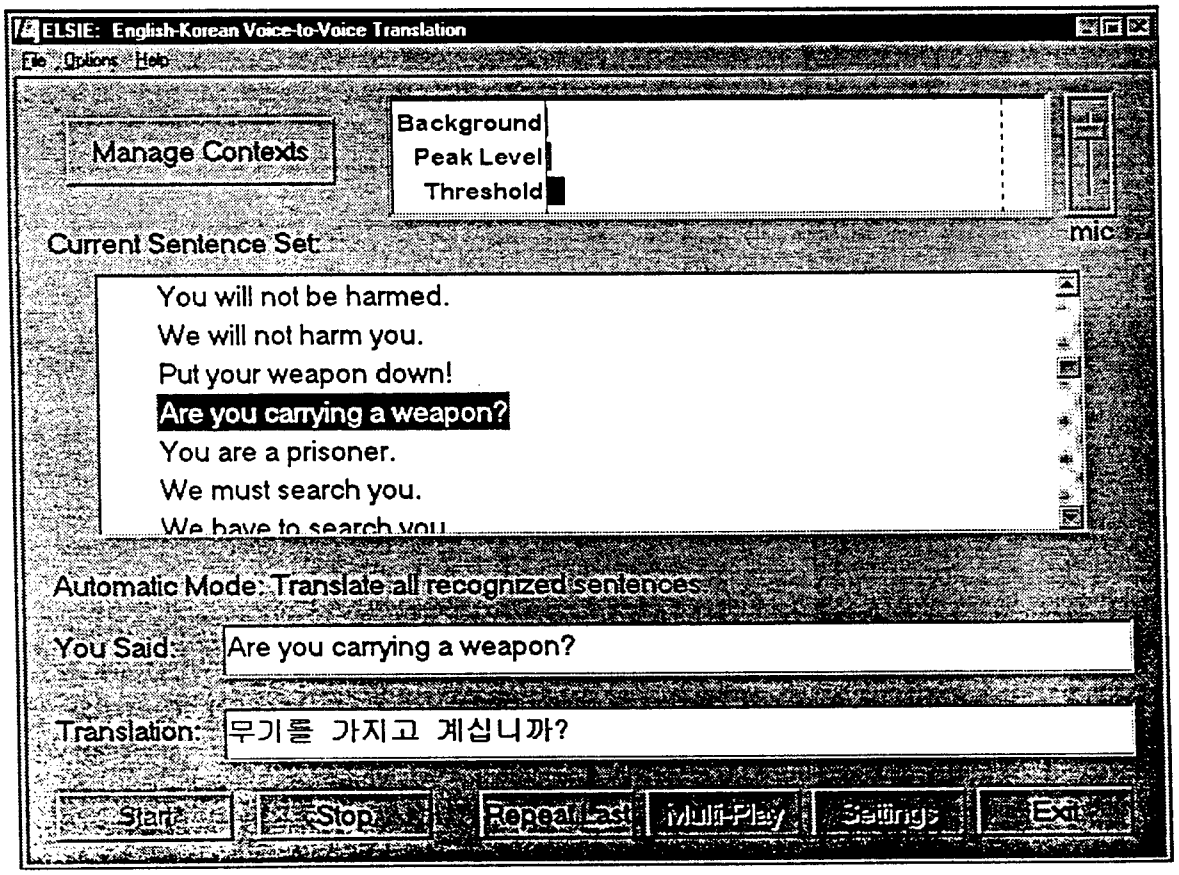
In this new version of the system (Version 1.92), the translation between English and Korean is performed through the normal translation path, in the same way as translation is accomplished for any other language pairs in our system (e.g. English-Spanish, English-Mandarin, English-French). The input English or Korean sentences go through various processing stages, including lexical look up, syntactic parse, and transfer. A detailed description of these processing stages is given in Milestone Report 23. To be consistent with the treatment of other languages in the system (especially with other Asian languages such Mandarin), the internal representation of Korean sentences during the translation process uses the romanized version. This is possible because of our system's new capability for dual representation of Korean sentences, described below.

3) Dual representation and display of Korean sentences

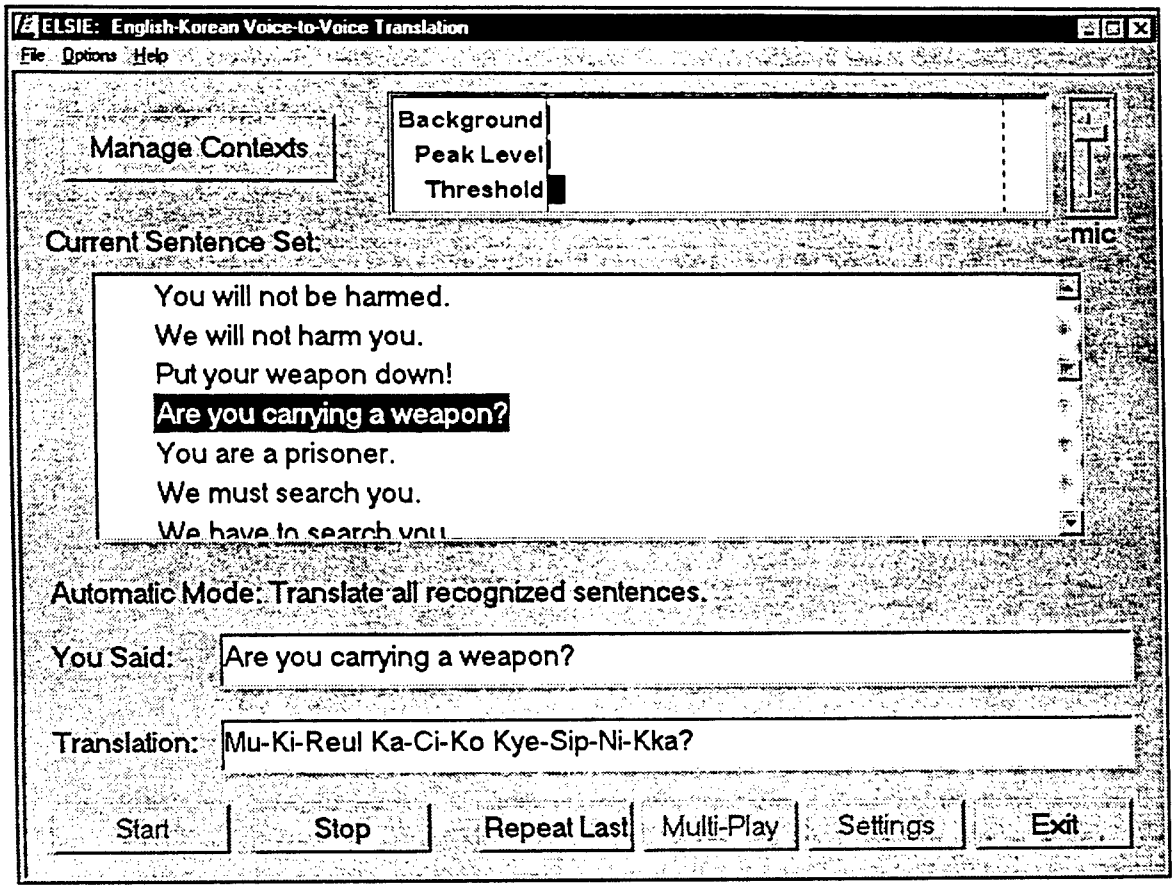
In the previous version, we could only display Korean sentences using Hangul (traditional Korean script), and there was no internal representation of Korean sentences

during the translation process, since English-to-Korean translation at the time did not go through the normal translation path.

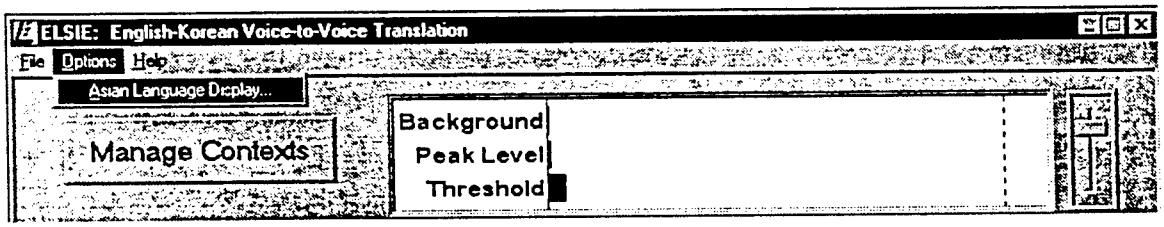
During the development of this new version of QRSLT/ELSIE, we utilized a public domain application called "hcode", which takes a file of Hangul script and produces a romanized version of the file, or vice versa. (The standard romanized system for Korean is called the McCune-Reischauer system.) The Korean data we developed (described above) was originally in Hangul script, since our Korean informants are not familiar with the McCune-Reischauer system (in fact, most Korean native speakers are unfamiliar with it). We then used the hcode application to produce the romanized version of these sentences to be used by the system internally. The Hangul version is used to display the Korean translations for native speakers, or others who can read Hangul, as illustrated below.



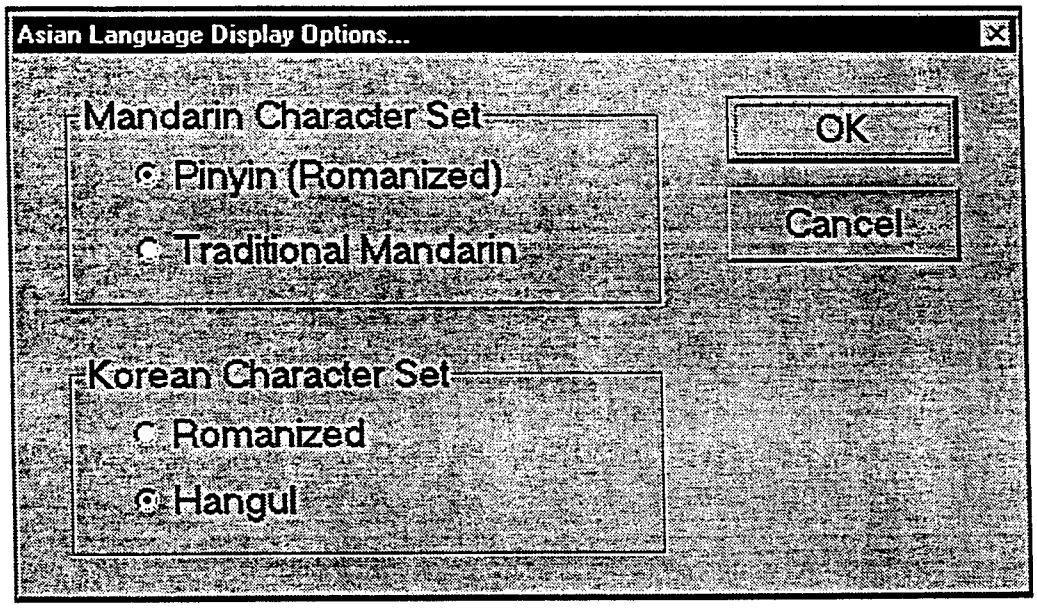
For the convenience of English-speaking users who cannot read Hangul, but nonetheless want to be able to know roughly how a Korean sentence is pronounced, we also built in the choice of displaying Korean output in the romanized version.



The choice between the Hangul or romanized display of Korean sentences can be made conveniently through a menu item selection displayed on the user interface.



Clicking on the Asian Language Display option brings up the following screen:



4) Korean speech recognizer

To provide a two-way translation capability that would allow processing of Korean responses to English questions, a Korean speech recognizer was required. Since there was none readily available to us, we bootstrapped a limited Korean recognition capability from the English recognizer in IBM ViaVoice. This basically involved building a phonetic Korean word dictionary for ViaVoice. The entries in this dictionary consist of two parts: the Korean word (in the romanized version), and its phonetic definition, represented as a series of phones derived from the inventory of phones used by ViaVoice to represent sound segments. The following is a small section of the dictionary that we built for the words in the Korean military and medical sentence sets:

a-pheu-se-yo AA M H AX S EH Y AO
a-pheu-sip-ni-kka AA M H AX SH IH P N IH G AA
chi-ryo-reul CH IH R Y AO R AX L
cu-se-yo JH UW S EH Y AO
eo-ti-e AX D IH EH
eo-ti-iss-seup-ni-kka AX D IH IH S S AX P N IH G AA
eo-ti-ka AX D IH G AA
eo-tteo-sip-ni-kka AX D T AX SH IH P N IH G AA
eum-cik-i-ci AX M CH IH K IH CH IH
hwan-ca-i-sip-ni-kka H W AA N CH AA IH SH IH P N IH G AA
iss-eu-sip-ni-kka IH S AX SH IH P N IH G AA
iss-seup-ni-kka IH S S AX P N IH G AA
ka-kil K AA G IH L
ka-seum-i K AA S AX M IH
ka-syeo-ya-kess-seup-ni-ta K AA S Y AX Y AA G EH S S AX P N IH D AA
ku-keup-cha-reul K UW G AX P CH AA R AX L
ma-se-yo M AA S EH Y AO

Such phonetic dictionaries can be built manually, but it would be a very slow and error prone process. In order to automate this task, we have developed a table of Korean phones and corresponding Via Voice phonetic representations, and written several Perl scripts to prepare a unique word list from a data set and to use the phone table to automatically generate the phonetic dictionary for a list of input Korean words.

The following is the phone table that we have constructed. The first field in each line is a letter or letter sequence resenting a Korean word segment; the second field is the ViaVoice phonetic representation for the sound of that letter or letter sequence.

```
# Vowels
a       AA
e       EH
i       IH
o       AO
u       UW
ae      AE
eo      AX
eu      AX
oe      W AE
ya      Y AA
yae     Y AE
ye      Y AE
yeo     Y AX
yo      Y AO
yi      Y IH
yu      Y UW
wa      W AA
wae     W AE
we      W EH
weo     W AX
wi      W IY

# Consonants
b       B
c       JH
cc      JH
ch      CH
d       D
g       G
h       H
k       K
kk      K
l       L
m       M
```

n	N
ng	NG
p	P
r	R
s	S
sh	SH
ss	S
t	T
th	T

The correspondence between a Korean letter (or a letter sequence) and its pronunciation is fairly regular, which facilitated our construction of the phones table. However, there is a complicating factor, which is the fact that a Korean letter (or letter sequence) may be pronounced differently in different contexts. For example, according to our Korean phrasebook, *kk* is pronounced in different ways depending on its position within a word. It is pronounced *kk* in initial position, *k* in final position, and *g* in the "middle" position (ending a syllable but followed by another syllable starting with a consonant in the same word). No phonetically detailed information is given on these environments (e.g., *g* would be expected if the preceding syllable ends in a voiced consonant, but would be phonetically unlikely if the preceding syllable ends in a voiceless consonant), so further refinement of the phone table will be necessary when additional resources are available for more in-depth analysis of Korean phonology.

The contextual variation is handled by the Perl script that creates phonetic dictionaries for input word lists. The Perl script makes use of the phone table, but has rules to resolve the issues of contextual variation.

Using the phonetic dictionary constructed for the words in a set of Korean sentences, ViaVoice can recognize the Korean sentences as if it were recognizing English sentences.

The accuracy of this bootstrapped Korean recognizer is about 80%. It would not be easy to improve this, simply because there are many sounds in Korean which do not exist in English. When we build the phonetic dictionary for Korean words, we are limited to using the phones for English sounds, which are used by the ViaVoice English recognizer.

5) Korean speech generation

As in the case of Korean speech recognition, no Korean synthesizer was readily available for this development. Thus it was necessary to generate Korean output speech via digital audio playback. To this end, several versions of the Korean phrases and sentences in both the military and medical sentence sets were prerecorded by a native speaker of Korean, and the resulting wave files were used to generate Korean output.

*U.S. GOVERNMENT PRINTING OFFICE: 2000-510-079-10007

DISTRIBUTION LIST

addresses	number of copies
AFRL/IFED 32 BROOKS ROAD ROME, NY 13441-4114	20
LANGUAGE SYSTEMS INC. 5959 TOPANGA CANYON BLVD. SUITE 340 WOODLAND, CA 91367-3648	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	1
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P.STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

AFRL/HESC-TDC 1
2698 G STREET, BLDG 190
WRIGHT-PATTERSON AFB OH 45433-7604

ATTN: SMDC IM PL 1
US ARMY SPACE & MISSILE DEF CMD
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

TECHNICAL LIBRARY D0274(PL-TS) 1
SPAWARSYSCEN
53560 HULL ST.
SAN DIEGO CA 92152-5001

COMMANDER, CODE 4TL000D 1
TECHNICAL LIBRARY, NAWC-WD
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6100

CDR, US ARMY AVIATION & MISSILE CMD 2
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RD-OB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000

REPORT LIBRARY 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

ATTN: D'BORAH HART 1
AVIATION BRANCH SVC 122.10
FOB10A, RM 931
300 INDEPENDENCE AVE, SW
WASHINGTON DC 20591

AFIWC/MSY 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

ATTN: KAROLA M. YOURISON 1
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE
PITTSBURGH PA 15213

USAF/AIR FORCE RESEARCH LABORATORY 1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB MA 01731-3004

ATTN: EILEEN LADUKE/D460 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

OUSD(P)/DTSA/DUTD 1
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

ELOQUENT TECHNOLOGY, INC. 1
2389 NORTH TRIPHAMMER ROAD
ITHACA, NY 14850

NLECTC-NORTHEAST 5
26 ELECTRONIC PARKWAY
ROME, NY 13441

AFRL/IFEC 5
C/O DR. BENINCASA
32 BROOKS ROAD
ROME, NY 13441-4114

OLECTC 2
WHEELING JESUIT UNIVERSITY
316 WASHINGTON AVE
WHEELING, WV 26003

BORDER RESEARCH AND TECH CTR 2
225 BROADWAY, SUITE 740
SAN DIEGO, CA 92101

MISSION OF AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.